



# 变换域下矩阵稀疏计算

高卫国<sup>1\*</sup>, 李颖洲<sup>1</sup>, 张思哲<sup>1</sup>, 蒋培刚<sup>2</sup>, 竺旭东<sup>2</sup>, 朱有团<sup>2</sup>

1. 复旦大学数学科学学院, 上海 200433;

2. 华为公司无线网络 RAN 算法部, 上海 201721

E-mail: wggao@fudan.edu.cn, yingzhouli@fudan.edu.cn, 18300180028@fudan.edu.cn, jiangpeigang@huawei.com, zhuxudong5@huawei.com, zhuyoutuan@huawei.com

收稿日期: 2024-04-30; 接受日期: 2025-04-15; 网络出版日期: 2025-10-14; \* 通信作者  
“十四五”国家重点研发计划“数学和应用研究”重点专项(批准号: 2021YFA1003305)资助项目

**摘要** 本文针对具有变换域下稀疏特征的矩阵簇运算提出一个新的快速计算框架, 从基本计算原理出发, 论述常见矩阵操作的实现机理及分拆策略, 以 Cholesky 分解、矩阵求逆及奇异值分解(singular value decomposition, SVD) 为例给出实现细节, 并探索近似和迭代算法在变换域下的可行性. 数值例子验证了所提框架下新型计算的正确性, 在列数较少及稀疏性较高的情况下性能获得较大提升, 为下一代无线通信等应用提供新的可能方案.

**关键词** 矩阵簇 变换域 Cholesky 分解 矩阵逆 SVD

**MSC (2020) 主题分类** 65F05, 65F15, 65F99

## 1 矩阵簇在变换域下的稀疏特征

很多实际应用问题中会碰到高维数据, 例如, 在无线通信中, 其上行信道的理想信道信息可看成一个空时频多维张量(如图 1 所示), 其中频域的最小粒度为子载波, 时域则对应不同时刻接收的信号, 空域对应基站侧的多天线阵子.

经常地, 给定信道集合  $H \in \mathbb{C}^{n \times m \times k}$  是一个三维张量, 其中  $n$  表示基站侧天线数,  $m$  表示用户侧天线数,  $k$  表示频域子载波数目,  $\{H(:, :, i)\}_{i=1}^k$  是  $k$  个  $n \times m$  复矩阵, 需要对矩阵簇(即这  $k$  个复矩阵) 执行同样的矩阵运算操作. 例如, 在进行信道权值计算时需要作奇异值分解(singular value decomposition, SVD), 就是计算这  $k$  个复矩阵的左奇异向量矩阵  $\{U(:, :, i)\}_{i=1}^k$ . 特别地, 称  $H$  为矩阵簇, 有别于一般的三维张量.

在单用户情形, 典型地有  $n = 64$ ,  $m = 4$ , 而  $k$  值相对要大得多, 可达 816 和 1632 等. 数据通常满足进一步的假设, 例如  $\{H(:, :, i)\}_{i=1}^k$  具有较强的连续相依性, 这时可以通过采样或者更新算法来提高

英文引用格式: Gao W G, Li Y Z, Zhang S Z, et al. Matrix computations exhibiting sparsity under domain transformation (in Chinese). Sci Sin Math, 2025, 55: 1–26, doi: 10.1360/SSM-2024-0143

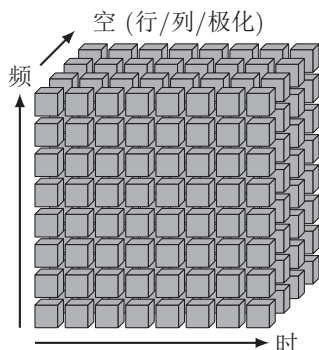


图 1 理想信道的多维张量表示

矩阵簇的计算效率, 但往往存在误差大和计算复杂度难控制等困难. 由于  $\{H(:, :, i)\}_{i=1}^k$  满足连续性且变化速度慢, 所以其满足变换域稀疏性假定, 即矩阵簇  $H$  在 Fourier 变换下具有稀疏性, 也就是  $nm$  个向量  $\{H(i, j, :)\}_{i=1, j=1}^{n, m}$  的 Fourier 系数仅在  $p$  个低频位置较大, 其余频率位置较小可忽略, 并且  $k \gg p$ , 如图 2 所示, 其中阴影部分表示非零元, 带  $\hat{\cdot}$  的变量表示 Fourier 变换后的系数.

变换域下稀疏性假设在很多时变相关的应用中满足, 包括气象和雷达等, 非零元位置可能有所不同, 只要位置连续或者间隔固定不会造成分析上的本质区别. 在这一前提下, 同时实现  $k$  组矩阵向量乘积、矩阵与矩阵乘积、SVD、特征值分解、求逆、广义逆、迫零等常用运算. 因此, 如何利用时域系数稀疏性实现频域矩阵操作的高效快速运算是一个重要而且有应用背景的问题.

我们将变换域稀疏计算问题表述为如下形式: 一般地, 对于矩阵簇  $H$  和矩阵运算  $f$  (如 QR 分解和 SVD 等), 记  $f(H)$  为  $\{f(H(:, :, i))\}_{i=1}^k$  组成的矩阵簇, 则称

$$g: \hat{H} \mapsto \widehat{f(H)}$$

为对应的变换域计算. 在变换域稀疏性假定下, 我们希望变换域计算  $g$  的复杂度低于逐个计算得到  $f(H)$ , 此时在以下两种应用场景中, 变换域计算比原始域计算更高效:

- (1) 结果作为  $f(H)$  的稀疏表示直接使用, 例如, 在变换域中用于此后的进一步计算;
- (2) 返回原始域计算出结果 (由于实际应用中域变换往往有高效的硬件实现, 所以反变换的复杂度将不被考虑).

从张量分析的角度考虑, 进行域变换也已成为一种常用技巧. 例如, 文献 [4] 基于 FFT (fast Fourier transform) 将矩阵运算推广为三维张量的运算, 由此定义了张量的  $t$ -乘积和  $t$ -SVD 等运算, 并进一步将这种定义方式推广到一般的可逆线性变换 [3], 这些操作在张量压缩、图像处理和数据补全等领域都

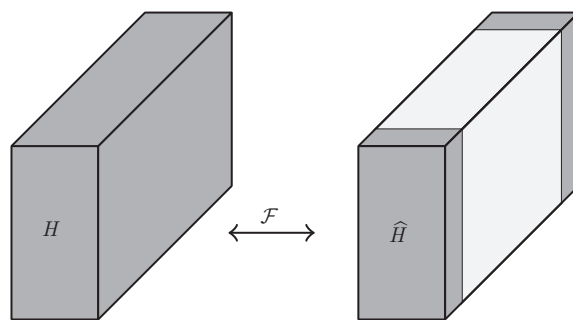


图 2 矩阵簇的变换域稀疏性

有应用. 其定义的  $t$ -乘积和  $t$ -SVD 等运算, 与本文中的变换域矩阵乘法、变换域矩阵 SVD 运算是一致的, 因此本文的问题也可理解为具有稀疏性的张量  $t$ -SVD 计算问题.

本文统称通过变换域稀疏性实现矩阵簇运算为矩阵稀疏计算, 有别于传统的稀疏矩阵计算.

## 2 矩阵稀疏计算原理及框架

这里仅关注 Fourier 变换下的稀疏计算, 为了记号方便, 分别用  $\mathcal{F}$  和  $\mathcal{F}^{-1}$  表示离散 Fourier 变换及其逆变换, 在算法表述中分别记为 FFT 和 IFFT. 对于向量  $x$ , 用  $p_x$  表示  $\hat{x}$  的非零元数目.

### 2.1 稀疏计算原理

为了探讨稀疏计算的可行性和实现方案, 需要从向量的基本四则运算开始. 首先列举两个基本事实:

**基本事实 2.1** 对于两个  $k$  维向量  $a$  和  $b$ , 加减得到的向量  $a \pm b$  的 Fourier 系数  $\widehat{a \pm b}$  满足

$$\widehat{a \pm b} = \hat{a} \pm \hat{b}.$$

**基本事实 2.2** 对于两个  $k$  维向量  $a$  和  $b$ , 逐元素相乘得到的向量  $a \circ b$  的 Fourier 系数  $\widehat{a \circ b}$  满足

$$\widehat{a \circ b} = \hat{a} * \hat{b},$$

其中  $\hat{a} * \hat{b}$  表示  $a$  的 Fourier 系数与  $b$  的 Fourier 系数的循环卷积<sup>1)</sup>, 见图 3(b).

由基本事实 2 可知, 若  $a$  和  $b$  的 Fourier 系数仅在首末  $p$  个位置非 0, 则  $a \circ b$  的 Fourier 系数至多有首末  $2p - 1$  个位置非 0. 又由基本事实 1 可知, Fourier 变换具有线性性质, 因此有如下定理:

**定理 2.1** 可分解为元素加、减、乘运算的连续矩阵簇计算均可实现 Fourier 变换下的稀疏计算, 其非零元个数与元素乘法运算的最长长度成正比.

值得指出的是, 对于非零元素长度分别为  $p_a$  和  $p_b$  的两个向量  $\hat{a}$  和  $\hat{b}$ , 加、减运算自然只需要对于非零部分进行计算, 但直接计算卷积需要  $O(p_a p_b)$  的运算量, 很可能超过传统稠密计算的  $k$  次运算, 为此, 我们通过快速 Fourier 变换进行转化, 可将卷积计算的复杂度降为  $O((p_a + p_b) \log(p_a + p_b))$ .

为了说明具体的算法, 先引入一些记号. 对于满足变换域稀疏性的向量  $\hat{x}$ , 设其末尾和开头分别有  $l_x$  和  $s_x$  个非零元, 用  $\hat{x}_j$  表示  $\hat{x}$  在  $j$  位置的元素 (从 0 开始标号), 当  $j$  不在 0 到  $k - 1$  范围内时, 规定  $\hat{x}_j = \hat{x}_{j \pm k}$ . 本文称向量  $\hat{x}$  的非零部分元素按  $\hat{x}_{-l_x}, \dots, \hat{x}_{s_x - 1}$  的顺序排列为自然顺序, 由此得到的长度为  $p_x$  的向量为  $\hat{x}_{nz}$ . 另外, 在算法中还需要在  $\hat{x}_{nz}$  后补上部分 0, 我们将  $\hat{x}_{nz}$  补零至长度为  $p'$  所得的向量记作  $\hat{x}_{nz+p'}$ .

考虑两个稀疏向量的卷积  $\hat{a} * \hat{b}$ , 其满足稀疏性假设  $p_a + p_b < k$ . 定义两个多项式

$$q_a(y) = \sum_{j=-l_a}^{s_a-1} a_j y^j, \quad q_b(y) = \sum_{j=-l_b}^{s_b-1} b_j y^j.$$

1) 为了避免卷积和乘积的对应关系中出现关于长度  $k$  的系数, 本文中 Fourier 变换定义为  $\hat{x} = \frac{1}{k} Fx$ , 逆变换为  $x = F^H \hat{x}$ ,  $F$  为 DFT (discrete Fourier transform) 矩阵, 其  $(i, j)$  元素为  $e^{2\pi\sqrt{-1}(i-1)(j-1)/k}$ .

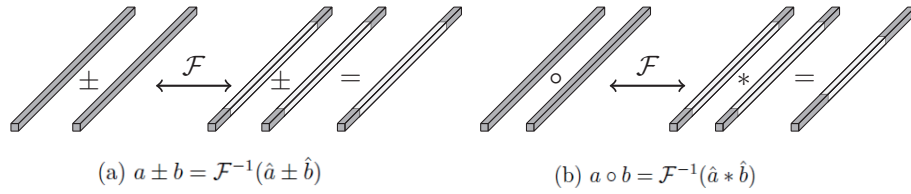


图 3 向量加减和点乘稀疏计算

根据 Fourier 变换的定义, 可得  $a_i = q_a(\zeta_i)$ ,  $b_i = q_b(\zeta_i)$ , 其中  $\zeta_i = e^{2\pi\sqrt{-1}i/k}$ . 记  $q = q_a q_b$  为乘积多项式, 则  $q$  的次数范围为  $-(l_a + l_b)$  到  $s_a + s_b - 2$ , 根据稀疏性假设, 成立

$$-(l_a + l_b) + k = k - (p_a - s_a) - (p_b - s_b) > s_a + s_b - 2,$$

因此可定义一个长度为  $k$  的向量  $\hat{x}$  使得  $\hat{x}(i)$  为多项式  $q$  中  $y^i$  项系数, 从而  $x_i = q(\zeta_i) = a_i b_i$ . 这说明  $\hat{x} = \hat{a} * \hat{b}$ , 且  $\hat{x}_{nz}$  正是  $q$  的系数组成的向量 (按升幂排序),  $p_x = p_a + p_b - 1$ . 另外考虑  $\hat{a}_{nz+p_x} * \hat{b}_{nz+p_x}$ , 根据卷积定义, 其  $i$  位置元素为

$$(\hat{a}_{nz+p_x} * \hat{b}_{nz+p_x})(i) = \sum_{j=0}^i \hat{a}_{nz+p_x}(j) \hat{b}_{nz+p_x}(i-j) + \sum_{j=i+1}^{p_x-1} \hat{a}_{nz+p_x}(j) \hat{b}_{nz+p_x}(i-j+p_x),$$

由于当  $j \leq p_a - 1$  且  $i - j + p_x \leq p_b - 1$  时, 才有  $\hat{a}_{nz+p_x}(j) \hat{b}_{nz+p_x}(i - j + p_x) \neq 0$ , 即要求  $p_a - 1 \geq j \geq p_x + i - p_b + 1$ , 但  $p_x + i - p_b + 1 \geq p_a$ , 因此该项恒为 0, 从而成立

$$(\hat{a}_{nz+p_x} * \hat{b}_{nz+p_x})(i) = \sum_{j=0}^i \hat{a}_{nz+p_x}(j) \hat{b}_{nz+p_x}(i-j) = \sum_{\substack{j_1+j_2=i \\ 0 \leq j_1 \leq p_a-1 \\ 0 \leq j_2 \leq p_b-1}} \hat{a}_{nz}(j_1) \hat{b}_{nz}(j_2),$$

即  $\hat{a}_{nz+p_x} * \hat{b}_{nz+p_x}$  为  $\hat{a}_{nz}$  和  $\hat{b}_{nz}$  进行多项式乘积所得, 而  $\hat{x}_{nz}$  也是两者进行多项式乘积所得, 因此成立

$$\hat{x}_{nz} = \hat{a}_{nz+p_x} * \hat{b}_{nz+p_x}.$$

因此, 根据卷积和乘积在 Fourier 变换下的关系, 得到  $\hat{x}_{nz} = \mathcal{F}[\mathcal{F}^{-1}(\hat{a}_{nz+p_x}) \circ \mathcal{F}^{-1}(\hat{b}_{nz+p_x})]$ , 这一过程除 FFT 外仅有长度为  $p_a + p_b - 1$  的向量点乘达到了我们期望的复杂度. 另外, 将上述过程中的  $p_x$  取为  $p' \geq p_x$ , 则还可以说明成立

$$\hat{x}_{nz+p'} = \mathcal{F}[\mathcal{F}^{-1}(\hat{a}_{nz+p'}) \circ \mathcal{F}^{-1}(\hat{b}_{nz+p'})],$$

后文将用到这一结论.

在我们的变换域稀疏性假设下, 向量按位置的加、减、乘运算都可以较好地保持原来向量的稀疏性, 但除法与开方运算一般而言没有这种性质. 对于向量按元素除法后的 Fourier 系数  $\hat{x} = \widehat{b/a}$ , 除了  $\hat{a}$  只有一个元素非零的平凡情形, 均不能保证有稀疏性, 我们用一个比较极端的例子说明: 取  $b$  为元素全部等于 1 的向量,  $\hat{a}(0) = 1$ ,  $\hat{a}(1) = -z$ ,  $|z| < 1$ , 其余位置全为 0, 则可以说明  $\hat{x}(j) = z^j / (1 - z^k)$ . 当  $|z|$  接近于 1 时,  $x$  所有元素的模长都十分接近, 几乎没有稀疏性.

在变换域算法中, 有时需要计算的量本身并不涉及除法, 但利用除法会减少运算量, 即通过  $\hat{x} = \widehat{b/a}$  计算  $\hat{x}$ , 但  $x$  由向量按元素的加、减、乘运算可以得到, 因此具有变换域稀疏性, 此时  $\hat{x}$  的稠密度

为  $p_x = p_b - p_a + 1$ , 对于这种特殊情形, 我们仍然可以利用 FFT 得到复杂度为  $O(p_b \log p_b)$  的稀疏算法. 根据稀疏乘法的结论, 有  $\hat{b}_{nz} = \mathcal{F}[\mathcal{F}^{-1}(\hat{x}_{nz+p_b}) \circ \mathcal{F}^{-1}(\hat{a}_{nz+p_b})]$ , 从中可以求解出

$$\hat{x}_{nz+p_b} = \mathcal{F}[\mathcal{F}^{-1}(\hat{b}_{nz}) ./ \mathcal{F}^{-1}(\hat{a}_{nz+p_b})].$$

对于向量按元素开方所得的向量  $\hat{x} = \sqrt{\hat{a}}$ , 即便假设其具有变换域稀疏性, 目前也没有较好的办法在  $O(p \log p)$  的复杂度以内计算. 如果与除法一样得到方程

$$\mathcal{F}^{-1}(\hat{x}_{nz+}) \circ \mathcal{F}^{-1}(\hat{x}_{nz+}) = \mathcal{F}^{-1}(\hat{a}_{nz}),$$

则求解所得  $\mathcal{F}(\hat{x}_{nz+})$  的每个元素有两种取值方式, 但显然并不是任意的选择都可以满足  $\hat{x}_{nz+}$  的补零形式, 目前也没有发现适合的选择方式.

总结以上分析可得到如下定理:

**定理 2.2** 向量  $a$  和  $b$  的加减乘运算以及结果具有变换域稀疏性的除法均可实现稀疏计算, 且复杂度不超过  $(p_a + p_b) \log(p_a + p_b)$ .

最后, 变换域的复共轭计算  $\hat{a}$  也可由  $\hat{a}$  按对应关系  $\hat{a}_0 = \bar{\hat{a}}_0$ ,  $\hat{a}_i = \bar{\hat{a}}_{k-i}$  ( $1 \leq i \leq k-1$ ) 直接得到.

## 2.2 基本矩阵运算的稀疏计算实现

根据第 2.1 小节中的基本事实, 容易得到一些基本矩阵运算的稀疏计算算法 (见算法 1). 例如, 对于具有上述稀疏性质的三维矩阵簇  $A$  和  $B$ , 其对应位置矩阵相乘得到的矩阵簇  $AB$  的 Fourier 系数也只在至多  $2p-1$  个位置非 0.

---

### 算法 1 矩阵乘法稀疏计算

---

输入:  $n \times m \times k$  稀疏张量  $\hat{A}$ ,  $m \times l \times k$  稀疏张量  $\hat{B}$

输出:  $n \times l \times k$  稀疏张量  $\hat{M}$ , 满足  $M(:, :, i) = A(:, :, i)B(:, :, i)$ ,  $1 \leq i \leq k$

```

1: for  $i = 1 : n$  do
2:   for  $j = 1 : l$  do
3:      $\hat{M}(i, j, :) = \sum_{s=1}^m \hat{A}(i, s, :) * \hat{B}(s, j, :)$ 
4:   end for
5: end for
```

---

例如, 对于  $m$  阶方阵  $A$ , 其行列式由以下公式定义:

$$\det(A) = \sum_{(i_1, i_2, \dots, i_m) \text{ 为全排列}} (-1)^{\mathcal{N}(i_1, i_2, \dots, i_m)} a_{1i_1} \cdots a_{mi_m},$$

其中  $\mathcal{N}(i_1, i_2, \dots, i_m)$  表示  $(i_1, i_2, \dots, i_m)$  的逆序数. 因此对于矩阵簇  $H \in \mathbb{C}^{m \times m \times k}$ , 其每个位置的矩阵行列式的 Fourier 系数可以通过稀疏计算得到, 只在至多  $mp$  个位置非 0. 相应算法见算法 2.

---

### 算法 2 行列式稀疏计算

---

输入:  $m \times m \times k$  稀疏张量  $\hat{C}$

输出:  $k$  维稀疏向量  $\hat{d}$  使得  $d(i) = \det(C(:, :, i))$

```

1:  $\hat{d} = \sum_{(i_1, i_2, \dots, i_m) \text{ 为全排列}} (-1)^{\mathcal{N}(i_1, i_2, \dots, i_m)} \hat{C}(1, i_1, :) * \hat{C}(2, i_2, :) * \cdots * \hat{C}(m, i_m, :)$ 
```

---

综上, 有如下推论:

**推论 2.1** 矩阵乘积 ( $C = AB$ )、协方差 ( $C = A^H A$ )、行列式 ( $|A|$ ) 等运算可稀疏计算.

当然, 直接用定义来计算  $m$  阶矩阵行列式的运算量为  $O(m!)$ , 当  $m$  不大 (如  $m \leq 4$ ) 时, 运算量可以接受; 而当  $m$  较大时, 要考虑采用其他算法, 如 LU 分解等  $O(m^3)$  复杂度的算法, 但相关算法涉及除法或者开方运算, 向量进行这些运算后并不像乘法一样可以保持变换域稀疏性, 具体的实现方式将在之后讨论.

### 2.3 稀疏计算基本框架

一般地, 对于满足变换域稀疏性的单个或多个矩阵簇,  $f$  为矩阵运算且可表达为矩阵元素的基本运算 (定理 2.2 中的运算) 的组合, 则相对于逐个计算得到  $f(A)$  的常规方法, 变换域中  $\widehat{f(A)}$  的计算可通过等价稀疏运算得到. 算法的基本框架如图 4 所示.

需要指出的是, 其他常用矩阵运算, 如 QR 分解和特征值分解等, 涉及加、减、乘、除、开方等运算, 对于除法和开方运算也可以通过合理分拆实现稀疏计算. 一般地, 假设  $g$  是任意的矩阵运算, 将  $g$  拆分为  $g = \phi(f_1, \dots, f_m)$ , 各个  $f_i$  只涉及定理 2.2 中的基本运算, 则所有  $f_i$  可进行变换域稀疏计算, 我们将这一组结果作为  $g$  的稀疏表示, 必要时回到原始域逐个矩阵通过  $\phi$  组合所需结果  $g$ . 例如对于矩阵求逆问题, 可以拆分为  $A^{-1} = A^* / \det(A)$ , 其中  $A^*$  为伴随矩阵, 其每个元素均为  $A$  的子矩阵的行列式, 因此两部分均可由变换域计算得到.

### 2.4 稀疏计算的误差分析

在实际应用中, 稀疏性是通过将原向量 Fourier 系数较小的值截断而得到的, 这一过程会导致结果产生一定的误差. 由于 Fourier 变换域是正交变换, 所以考虑基于 Frobenius 范数的误差分析, 假设对原向量  $a$  和  $b$  的 Fourier 系数截去的向量为  $\delta\hat{a}$  和  $\delta\hat{b}$ , 造成的原向量的变化为  $\delta a$  和  $\delta b$ , 则有

$$\|\delta a\|_F = \|\delta\hat{a}\|_F, \quad \|\delta b\|_F = \|\delta\hat{b}\|_F.$$

由此通过简单的数学推导可以得到

$$\|\delta(a \pm b)\|_F^2 \leq 2(\|\delta\hat{a}\|_F^2 + \|\delta\hat{b}\|_F^2),$$

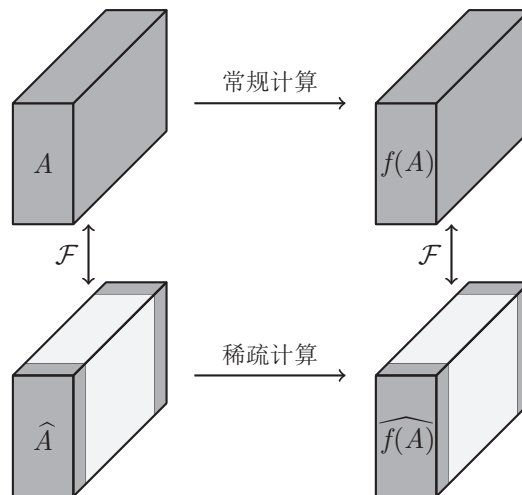


图 4 稀疏计算框架

$$\|\delta(a \circ b)\|_1 \leq \|a\|_F \|\delta \hat{b}\|_F + \|b\|_F \|\delta \hat{a}\|_F + \|\delta \hat{a}\|_F \|\delta \hat{b}\|_F.$$

由此可见, 只要截去的能量相对于原来的能量较小, 进行稀疏计算得到的相对误差也就较小, 截断的影响并不会随着运算次数的增加而增加. 因此在实际计算中, 只要向量的 Fourier 系数中有较大一部分相对于其余系数模长较小, 就可以认为其具有稀疏性, 并应用各类稀疏计算的算法.

重要的是, 传统的基于范数的误差分析也适用于稀疏计算. 对于矩阵簇  $A \in \mathbb{C}^{n \times m \times k}$ , 记

$$\|A\|_F = \sqrt{\sum_{i=1}^n \sum_{j=1}^m \|A(i, j, :)\|_F^2}.$$

由于 Fourier 变换保持每个  $\|A(i, j, :)\|_F$  不变, 因此对于整个矩阵簇也有  $\|\delta A\|_F = \|\delta \hat{A}\|_F$ . 以求逆问题为例, 根据传统的误差分析结果, 当  $\|\delta A_i\|_F / \|A_i\|_F \leq 1/(2\kappa_F(A_i))$  时, 成立

$$\frac{\|\delta A_i^{-1}\|_F}{\|A_i^{-1}\|_F} \leq 2\kappa_F(A_i) \frac{\|\delta A_i\|_F}{\|A_i\|_F},$$

记  $K = \max_i \kappa_F(A_i)$ ,  $C = \|A\|_F / (\sqrt{k} \min_i \|A_i\|_F)$ . 由于变换域计算对于截断的非零部分是准确的, 因此根据 Fourier 变换的正交性, 成立

$$\frac{1}{k} \sum_i \frac{\|\delta A_i^{-1}\|_F}{\|A_i^{-1}\|_F} \leq 2k^{-1} K \|\delta \hat{A}\|_F \sqrt{\sum_i \frac{1}{\|A_i\|_F^2}} = 2KC \frac{\|\delta \hat{A}\|_F}{\|A\|_F},$$

即在矩阵簇的各个矩阵条件数不太坏、范数比较接近时, 只要截断部分的相对能量较小, 矩阵求逆的平均的相对误差也较小. 一般地, 对各类矩阵计算问题均可类似分析稀疏截断带来的误差.

### 3 一般矩阵运算的稀疏计算及实现

#### 3.1 LU 分解与 Cholesky 分解

LU 分解 (Cholesky 分解) 是非奇异 (对称正定) 矩阵的一个基本运算, 可用于求解线性方程组等问题, 在无线通信问题中也有应用. 传统的 LU 分解 (Cholesky 分解) 每一步消去需要用到除法 (以及开方), 难以与基于 FFT 的稀疏计算直接结合, 因此需要对其改造并进行拆分计算.

首先考察传统的 LU 分解:

$$A_0 = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} 1 & & & \\ l_{21} & 1 & & \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \cdots & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ & u_{22} & \cdots & u_{2n} \\ & & \ddots & \vdots \\ & & & u_{nn} \end{bmatrix},$$

重写为  $2 \times 2$  分块形式为

$$\begin{bmatrix} a_{11} & a_{1:} \\ a_{:1} & \tilde{A}_1 \end{bmatrix} = \begin{bmatrix} 1 & \\ l_1 & L_1 \end{bmatrix} \begin{bmatrix} u_{11} & u_1^H \\ & U_1 \end{bmatrix},$$

其中

$$l_1 = a_{:1}/a_{11}, \quad u_1 = a_{1:}^H,$$

并且

$$L_1 U_1 = \tilde{A}_1 - a_{:1} a_1 / a_{11} := A_1,$$

因此算法可以通过循环/递归实现, 但在运算过程中要用到除法, 不适合直接进行稀疏计算. 一个自然的解决方案是, 采用拆分框架, 将分子、分母分开计算, 则只涉及加减乘运算. 考虑第  $i$  步的分解:

$$A_{i-1} = \begin{bmatrix} a_{ii} & a_{i:} \\ a_{:i} & \tilde{A}_i \end{bmatrix} = \begin{bmatrix} 1 \\ l_i & L_i \end{bmatrix} \begin{bmatrix} u_{ii} & u_i^H \\ & U_i \end{bmatrix},$$

注意到如果  $A_{i-1}$  的所有元素分母相同, 则  $A_i = \tilde{A}_i - a_{:i} a_i / a_{ii}$  也有公共的分母, 而最初进行分解的矩阵  $A_0$  的元素分母均为 1, 因此在递归中不妨假设  $A_i$  的元素有公共的分母. 记  $A_{i-1}$  的分母为  $d_{i-1}$ , 分子记为  $A_{i-1}^{\text{nu}}$ , 则可以得到

$$\begin{aligned} l_i &= \frac{a_{:i}^{\text{nu}}}{a_{ii}^{\text{nu}} d_{i-1}}, \\ u_i^H &= \frac{a_{i:}^{\text{nu}}}{d_{i-1}}, \\ A_i &= \frac{\tilde{A}_i^{\text{nu}} a_{ii}^{\text{nu}} - a_{:i}^{\text{nu}} a_{i:}^{\text{nu}}}{a_{ii}^{\text{nu}} d_{i-1}}. \end{aligned} \quad (3.1)$$

由此可以通过递归/循环得到变换域的 LU 分解算法. 但在递归中, 如果记原矩阵  $A_{i-1}$  的分子分母的稠密度分别为  $p_n^{(i-1)}$  和  $p_d^{(i-1)}$ , 则下一步进行分解的矩阵  $A_i$  的分子分母稠密度分别为

$$p_n^{(i)} = 2p_n^{(i-1)} - 1, \quad p_d^{(i)} = p_n^{(i-1)} + p_d^{(i-1)} - 1.$$

设原始矩阵的分子稠密度为  $p$ , 分母稠密度为 1, 则可得到

$$p_n^{(i)} = 2^i(p - 1) + 1, \quad p_d^{(i)} = (2^i - 1)(p - 1) + 1,$$

而对一个  $n$  阶矩阵完成 LU 分解需要迭代  $n - 1$  次, 因此按照上述拆分框架得到的变换域算法复杂度超过  $2^n p$  量级, 这显然是无法接受的.

为了解决稠密度指数增长的问题, 需要 LU 分解中 Schur 补的分母  $d$  的具体形式. 仍将原矩阵  $A_0$  的 LU 分解写成  $2 \times 2$  的分块:

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} L_{11} & \\ & L_{22} \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ & U_{22} \end{bmatrix} = \begin{bmatrix} L_{11} & \\ & I \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ & A_i \end{bmatrix} =: \tilde{L} \tilde{U}, \quad (3.2)$$

其中  $A_{11}$  为  $i \times i$  的子矩阵,  $L_{22} U_{22} = A_i$ . 根据  $\tilde{L}$  的结构,  $\tilde{L} \tilde{U}$  为  $\tilde{U}$  经一系列初等的行变换所得, 也就是按一定顺序将第  $j$  行乘以系数加到下方的每一行上,  $1 \leq j \leq i$ , 因此对任意的  $s, t > i$ , 第  $1, \dots, i, s$  行、 $1, \dots, i, t$  列组成的  $\tilde{U}$  的子矩阵的行列式没有改变. 记上述行列式值为  $D_{st}^i$ ,  $a_{i,st}$  为  $A_i$  在  $(s, t)$  位置的元素. 由于  $\tilde{U}$  相应位置的子矩阵是一个上三角矩阵, 容易得到其行列式为  $\det(U_{11}) a_{i,st}$ . 另外有

$$A_{11} = L_{11} U_{11}, \quad \det(L_{11}) = 1,$$

从而得到

$$a_{i,st} = \frac{D_{st}^i}{|A_{11}|}.$$

这说明  $A_i$  的分母部分应为  $d_i = |A_{11}|$ , 其稠密度应为  $i(p-1)+1$  而非  $(2^i-1)(p-1)+1$ ; 另外取  $s=t=i+1$ , 则得到  $a_{i+1,i+1}$  的分子部分为  $D_{i+1,i+1}^i = |A_{i+1,i+1}| = d_{i+1}$ . 因此  $d_i = a_{ii}^{uu}$ , 从而对于 (3.1) 中的更新公式, 实质上的分子为  $(\tilde{A}_i^{uu} a_{ii}^{uu} - a_{:i}^{uu} a_{i:}^{uu})/d_{i-1}$ , 这里的除法结果本质上仍为加减乘运算所得, 因此适合变换域计算.

总结以上拆分方法, 按矩阵形式变换域的 LU 分解  $A = LU$  可表示为  $L = L_f D_L^{-1}$ ,  $U = D_U^{-1} U_f$ , 其中  $L_f$  为下三角矩阵,  $U_f$  为上三角矩阵,  $D_U$  和  $D_L$  为对角矩阵. 各部分均由变换域稀疏计算得到, 具体计算过程在算法 3 中, 其中  $*$  表示变换域乘法,  $\odot$  表示变换域的稀疏除法.

---

### 算法 3 矩阵 LU 分解变换域计算

---

输入:  $n \times n \times k$  三维张量  $\hat{A}$

输出:  $n \times n \times k$  三维张量  $\hat{L}_f, \hat{D}_L, \hat{D}_U, \hat{U}_f$

```

1: 设置  $\hat{D}_U(1,1,1) = 1$ , 其余均为 0
2: for  $i = 1 : n$  do
3:    $\hat{D}_L(i, i, :) = \hat{A}(i, i, :)$ 
4:   if  $i + 1 \leq n$  then
5:      $\hat{D}_U(i + 1, i + 1, :) = \hat{A}(i, i, :)$ 
6:   end if
7:   for  $j = i : n$  do
8:      $\hat{L}_f(j, i, :) = \hat{A}(j, i, :)$ 
9:      $\hat{U}_f(i, j, :) = \hat{A}(i, j, :)$ 
10:  end for
11:  for  $s = i + 1 : n$  do
12:    for  $t = i + 1 : n$  do
13:       $\hat{A}(s, t, :) = (\hat{A}(i, i, :) * \hat{A}(s, t, :) - \hat{A}(s, i, :) * \hat{A}(i, t, :)) \odot \hat{D}_U(i, i, :)$ 
14:    end for
15:  end for
16: end for

```

---

从算法 3 中  $L_f$  和  $U_f$  的赋值和  $A$  的更新中容易看出, 如果原矩阵  $A$  为 Hermite 矩阵, 则每次更新得到的  $A$  也为 Hermite 矩阵, 且  $U_f = L_f^H$ , 因此用上述算法计算得到  $A = L_f D_L^{-1} D_U^{-1} L_f^H$ , 从而 Cholesky 分解  $A = LL^H$  的拆分形式为  $L = L_f (D_L D_U)^{-1/2}$ , 相关的算法只需要将算法 3 中的  $U_f$  去掉, 并且只对  $s \geq t$  的情形更新  $\hat{A}(s, t, :)$  即可. 对于  $s < t$  的取值, 由变换域共轭即可得到.

另外, 由于  $U$  的对角元  $u_{ii}$  直接从  $A_i$  中得到, 而在前面的分析中我们已经得到  $A_i$  的  $(i+1, i+1)$  位置元素为  $d_{i+1}/d_i$ , 因此

$$u_{ii} = d_i/d_{i-1}, \quad d_0 = 1,$$

从而行列式  $|A| = d_n = u_{nn}^{uu}$ , 因此上述算法也可作为行列式的变换域计算方法.

## 3.2 矩阵求逆

在无线通信中, 常常也需要计算矩阵的逆, 传统的矩阵算法基于 Gauss 消元法, 相当于先计算矩阵的 LU 分解  $A = LU$ , 通过初等行变换将增广矩阵  $[L \ I_n]$  左侧的矩阵变为  $I$ , 右边的矩阵即得到  $L^{-1}$ ; 再通过初等行变换将  $[U \ L^{-1}]$  左侧的矩阵变为  $I$ , 得到的右侧矩阵即为  $A^{-1}$ . 由于相关的运算与 LU 分解一样只涉及加减乘除运算, 因此也自然地拆分为分子、分母两部分进行计算.

首先考虑  $L^{-1}$  的计算, 利用 Gauss 消元法计算  $L^{-1}$  所需要的行变换可分为  $n-1$  步, 其中第  $i$  步是对  $L$  的第  $i$  列, 用  $L$  的对角元消去对角线以下的部分. 为了更好地描述行变换对右侧单位阵的作

用, 并分析每一步中分子、分母的计算, 我们沿用此前的记号, 用  $l_i$  表示  $L$  第  $i$  列对角元以下的部分, 并记

$$L_i = \begin{bmatrix} I_{i-1} & & & \\ & 1 & & \\ & & \ddots & \\ & & & l_i & 1 \end{bmatrix},$$

则  $L = L_1 L_2 \cdots L_{n-1}$ , 且 (3.2) 中的  $\tilde{L} = L_1 \cdots L_i$ . 另外, 计算  $L^{-1}$  的过程相当于  $L^{-1} = L_{n-1}^{-1} \cdots L_1^{-1}$  从右向左依次进行矩阵乘法.

接下来讨论在计算过程中右侧矩阵的分子、分母变化情况. 假设  $i$  步消元已经完成, 则此时右侧矩阵为  $L_i^{-1} \cdots L_1^{-1} = \tilde{L}^{-1}$ . 根据 (3.2) 中的分块, 有  $A_{11} = L_{11} U_{11}$ ,  $A_{11} = L_{11} U_{11}$ ,  $A_{12} = L_{12} U_{11}$ , 另外  $\tilde{L}^{-1}$  的分块形式为

$$\tilde{L}^{-1} = \begin{bmatrix} L_{11}^{-1} & \\ -L_{21} L_{11}^{-1} & I \end{bmatrix}.$$

由于下一步对  $i+1$  列进行消元, 只涉及第  $i+1$  行到最后一行, 而  $L_{11}^{-1}$  的部分不变, 因此只需要考虑  $-L_{21} L_{11}^{-1}$  的分母部分. 由于

$$-L_{21} L_{11}^{-1} = -A_{12} U_{11}^{-1} L_{11}^{-1} = -A_{12} A_{11}^{-1},$$

因此根据伴随矩阵的性质, 其分母部分为  $d_i = |A_{11}|$ .

将 Gauss 消元过程中右侧的矩阵的分子部分记为  $K^{\text{nu}}$ , 则  $i$  步消元后, 右侧矩阵的第  $i+1$  到  $n$  行形如

$$\begin{bmatrix} \frac{k_{i+1,1}^{\text{nu}}}{d_i} & \cdots & \frac{k_{i+1,i}^{\text{nu}}}{d_i} & 1 & \\ \vdots & \ddots & \vdots & \ddots & \\ \frac{k_{n,1}^{\text{nu}}}{d_i} & \cdots & \frac{k_{n,i}^{\text{nu}}}{d_i} & & 1 \end{bmatrix},$$

对所有  $s > i+1$ , 将第  $i+1$  行乘上  $-l_{s,i+1}^{\text{nu}}/d_{i+1}$  加到第  $s$  行上, 则此时  $K$  的更新为

$$k_{st} \leftarrow k_{st} - l_{s,i+1}^{\text{nu}} k_{i+1,t} = \frac{k_{st}^{\text{nu}} d_{i+1} - l_{s,i+1}^{\text{nu}} k_{i+1,t}^{\text{nu}}}{d_i d_{i+1}}, \quad 1 \leq t \leq i,$$

$$k_{st} \leftarrow -\frac{l_{st}^{\text{nu}}}{d_{i+1}}, \quad t = i+1.$$

根据对  $K$  的分母的分析, 更新后的分母应为  $d_{i+1}$ , 因此对  $k_{st}^{\text{nu}}$  的更新为

$$k_{st}^{\text{nu}} \leftarrow \frac{k_{st}^{\text{nu}} d_{i+1} - l_{s,i+1}^{\text{nu}} k_{i+1,t}^{\text{nu}}}{d_i}, \quad 1 \leq t \leq i,$$

$$k_{st}^{\text{nu}} \leftarrow -l_{st}^{\text{nu}}, \quad t = i+1,$$

其中的除法同样也有结果的稀疏性保证, 适合变换域计算.

通过对  $i = 0, 1, \dots, n-2$  进行上述更新即可得到变换域求  $L^{-1}$  的算法, 另外从算法过程中可以看出  $L^{-1}$  的第  $i$  行的分母为  $d_{i-1}$ , 与  $U$  第  $i$  行的分母相同, 因此变换域计算的结果可写成矩阵形式  $L^{-1} = D_U^{-1} K_f$ , 其中  $K_f$  为分子部分. 以上变换域三角矩阵求逆的方法总结为算法 4.

**算法 4** 三角矩阵求逆变换域计算

输入:  $n \times n \times k$  三维张量  $\hat{L}_f, \hat{D}_L, \hat{D}_U$

输出:  $n \times n \times k$  三维张量  $\hat{K}$

```

1: for  $i = 1 : n$  do
2:    $\hat{K}(i, i, :) = \hat{D}_U(i, i, :)$ 
3:   for  $j = i + 1 : n$  do
4:      $\hat{K}(j, i, :) = -\hat{L}_f(j, i, :)$ 
5:     for  $s = 1 : i - 1$  do
6:        $\hat{K}(j, s, :) = (\hat{K}(j, s, :) * \hat{D}_L(i, i, :) - \hat{K}(i, s, :) * \hat{L}_f(j, i, :)) \odot \hat{D}_U(i, i, :)$ 
7:     end for
8:   end for
9: end for

```

得到  $L^{-1}$  后, 继续对  $[U L^{-1}]$  使用 Gauss 消元法计算  $A^{-1}$ . 为了便于分析计算过程中的分子、分母部分, 采用的消元顺序如下: 从第 1 列开始, 将该列的对角元化为 1, 再用对角元消去该列对角线以上的部分. 仍使用 (3.2) 中的记号, 当对前  $i$  列消元完成后, 相当于对前  $i$  行左乘了矩阵  $U_{11}^{-1}$ , 因此  $L^{-1}$  中的前  $i$  行变为  $[A_{11}^{-1} \ 0]$ , 而左侧  $U$  的相应部分变为  $[I \ U_{11}^{-1}U_{12}]$ . 由于

$$U_{11}^{-1}U_{12} = U_{11}^{-1}L_{11}^{-1}A_{12} = A_{11}^{-1}A_{12},$$

因此  $U$  和  $L^{-1}$  前  $i$  行的分母部分均为  $d_i$ .

根据上述分析, 当完成  $i$  步消元后, 增广矩阵的前  $i + 1$  行、 $n + i + 1$  列如下, 其余列为 0:

$$\left[ \begin{array}{cccc|cccc} 1 & \frac{u_{1,i+1}^{nu}}{d_i} & \dots & \frac{u_{1,n}^{nu}}{d_i} & \frac{k_{i,1}^{nu}}{d_i} & \dots & \frac{k_{1,i}^{nu}}{d_i} & \\ & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \\ & & & & & & & \\ & & & & & & & \\ 1 & \frac{u_{i,i+1}^{nu}}{d_i} & \dots & \frac{u_{i,n}^{nu}}{d_i} & \frac{k_{i,1}^{nu}}{d_i} & \dots & \frac{k_{i,i}^{nu}}{d_i} & \\ & \frac{u_{i+1,i+1}^{nu}}{d_i} & \dots & \frac{u_{i+1,n}^{nu}}{d_i} & \frac{k_{i+1,1}^{nu}}{d_i} & \dots & \frac{k_{i+1,i}^{nu}}{d_i} & \frac{k_{i,i}^{nu}}{d_i} \end{array} \right].$$

考察进行第  $i + 1$  步消元时  $U$  和  $K$  的分子部分的变化, 首先将  $u_{i+1,i+1}$  元素变为 1, 这相当于对第  $i + 1$  行乘上  $d_i/d_{i+1}$ , 因此只是改变了  $i + 1$  行各个元素的分子. 然后将第  $i + 1$  列的所有非零元消去, 此时  $U$  的第  $i + 1$  列变为 0, 而对于  $1 \leq s \leq i$ , 非零部分的更新为

$$k_{st} \leftarrow k_{st} - u_{s,i+1}k_{i+1,t} = \frac{k_{st}^{nu}d_{i+1} - u_{s,i+1}^{nu}k_{i+1,t}^{nu}}{d_i d_{i+1}}, \quad 1 \leq t \leq i,$$

$$k_{st} \leftarrow \frac{-u_{s,i+1}}{d_{i+1}}, \quad t = i + 1,$$

$$u_{st} \leftarrow u_{st} - u_{s,i+1}u_{i+1,t} = \frac{u_{st}^{nu}d_{i+1} - u_{s,i+1}^{nu}u_{i+1,t}^{nu}}{d_i d_{i+1}}, \quad i + 1 < t \leq n.$$

由于这些元素的分子部分都应该为  $d_{i+1}$ , 因此分子部分的更新可与之前类似地通过变换域的稀疏除法得到.

对  $i = 0, 1, \dots, n$  进行上述更新, 最终可得到  $A^{-1} = K_f/|A|$  形式的逆矩阵, 其中  $K_f$  为分子部分, 分母部分为行列式. 以上变换域求逆的方法可总结为算法 5.

**算法 5** 利用 LU 分解的矩阵求逆变换域计算输入:  $n \times n \times k$  三维张量  $\hat{A}$ 输出:  $n \times n \times k$  三维张量  $\hat{K}, \hat{d}$ 

- 1: 计算 LU 分解的分子、分母得到  $\hat{L}_f, \hat{D}_L, \hat{D}_U, \hat{U}_f$ , 根据算法 3 计算
- 2: 计算  $L^{-1}$  的分子部分得到  $\hat{K}$ , 根据算法 4 计算
- 3: **for**  $i = 1 : n$  **do**
- 4:   **for**  $s = 1 : i - 1$  **do**
- 5:     **for**  $t = 1 : i - 1$  **do**
- 6:        $\hat{K}(s, t, :) = (\hat{K}(s, t, :) * \hat{D}_L(i, i, :) - \hat{K}(i, t) * \hat{U}(s, i, :)) \oslash \hat{D}_U(i, i, :)$
- 7:     **end for**
- 8:      $\hat{K}(s, i, :) = -\hat{U}(s, i, :)$
- 9:     **for**  $t = i + 1 : n$  **do**
- 10:        $\hat{U}(s, t, :) = (\hat{U}(s, t, :) * \hat{D}_L(i, i, :) - \hat{U}(i, t) * \hat{U}(s, i, :)) \oslash \hat{D}_U(i, i, :)$
- 11:     **end for**
- 12:   **end for**
- 13: **end for**
- 14:  $\hat{d} = \hat{D}_L(n, n, :)$

**3.3 QR 分解**

矩阵的 QR 分解也是一类有广泛应用的基本矩阵运算, 传统的单个矩阵 QR 分解算法有多种方式实现, 不同方法所得结果的精度 ( $Q$  的正交性) 也有所不同.

在常见的传统方法中, 基于旋转或 Householder 变换的 QR 分解是一种比较精确的方法 (参见文献 [2, 第 3.4 小节]), 但对于变换域计算, 这些变换涉及混合的开方和四则运算, 不适合稀疏计算, 而进行公式拆分也涉及嵌套的开方等复杂形式, 难以得到有效的算法.

本文的变换域 QR 算法采用基于 Cholesky 的 QR 分解作为原型加以改造. Cholesky QR 是一类高效但误差对矩阵条件数敏感的算法, 不过需要指出的是, 在变换域算法的误差分析中已经说明, 截断 Fourier 系数较小的部分的合理性需要矩阵是良好条件的才能保证截断 Fourier 系数较小的部分是合理的, 因此对于变换域算法, 采用条件数敏感的算法也合理.

对于一个  $n \times m$  阶的列满秩矩阵  $A$ , Cholesky QR 首先对  $A^H A$  计算 Cholesky 分解得到

$$A^H A = LL^H,$$

则  $A$  的 QR 分解为

$$Q = AL^{-H}, \quad R = L^H.$$

其中涉及的矩阵运算包括乘法、Cholesky 分解和三角矩阵求逆, 这些运算都已有相应的变换域算法, 因此组合这些方法即可得到变换域的 QR 算法.

我们使用先前变换域 Cholesky 分解算法和三角矩阵求逆算法中的记号, 描述变换域 QR 的拆分形式. 首先对  $A^H A$  进行 Cholesky 分解得到  $A^H A = L_f D_L^{-1} D_U^{-1} L_f^H$ , 因此得到

$$R = (D_L D_U)^{-1/2} L_f^H.$$

然后对三角矩阵  $L$  求逆并计算  $AL^{-H}$ , 由于变换域求逆的拆分形式为  $(L_f D_L^{-1})^{-1} = D_U^{-1} K_f$ , 因此

$$Q = A(L_f (D_U D_L)^{-1/2})^{-H} = AK_f^H (D_L D_U)^{-1/2}.$$

总结以上过程, 得到变换域的 QR 分解算法, 其拆分形式为

$$A = QR = (Q_f D^{-1/2})(D^{-1/2} R_f),$$

流程总结为算法 6.

---

**算法 6** 矩阵 QR 分解变换域计算

---

输入:  $n \times m \times k$  三维张量  $\hat{A}$

输出:  $n \times m \times k$  三维张量  $\hat{Q}_f$ ,  $m \times m \times k$  三维张量  $\hat{D}$  和  $\hat{R}_f$

- 1: 根据算法 1 计算  $\hat{H} = \hat{A}^H * \hat{A}$
  - 2: 根据算法 3 的 Cholesky 分解版本计算  $\hat{H} = \hat{L}_f * \hat{D}_L^{-1} * \hat{D}_U^{-1} * \hat{L}_f^H$
  - 3:  $\hat{R}_f = \hat{L}_f^H$ ,  $\hat{D} = \hat{D}_L * \hat{D}_U$
  - 4: 根据算法 4 计算  $(L_f D_L^{-1})^{-1} = D_U^{-1} K_f$
  - 5:  $\hat{Q}_f = \hat{A} * \hat{K}_f$
- 

### 3.4 低阶 SVD 计算

最后讨论矩阵的特征值和奇异值问题, 在无线通信的单用户问题中, 小阶矩阵的 SVD 是常见的运算, 其列数  $m$  一般满足  $m \leq 4$ , 对此我们给出具体的变换域算法, 在下一小节中再尝试将方法推广至一般的情形.

传统的 SVD 算法 (参见文献 [2, 第 5.3 小节]) 一般都涉及不适合变换域计算的操作, 例如, dqds 算法和分而治之法对矩阵进行二对角化的预先处理, 需要用到旋转或 Householder 变换, 二分法需要计算矩阵的惯性指数, 而惯性指数是整数, 不具有变换域的压缩性质. 因此, 根据变换域计算适合的运算和矩阵阶数较小的特点, 我们使用以下介绍的特征值法.

记  $A$  的奇异值分解为  $A = U \Sigma V^H$ , 则

$$U = A(\Sigma V^H)^{-1} = AV \Sigma^{-1},$$

而  $A^H A = V \Sigma^2 V^H$ , 即  $V$  为  $A^H A$  的特征向量, 而  $U$  在每列相差一个归一化系数的意义下等于  $AV$ . 因此, 我们将 SVD 的计算分解为矩阵乘法和对称特征值问题.

首先矩阵乘法  $H = A^H A$  可直接在变换域中实现. 对于矩阵  $H$  的特征向量的计算, 先通过特征多项式计算其特征值, 再通过求解奇异的线性方程组得到特征向量. 具体而言,  $H$  的特征多项式的系数可由其主子式之和得到, 因此只涉及加减乘运算, 之后根据求根公式, 将特征值的表达式涉及除法和开方运算的部分进行适当拆分, 变换域计算各个中间量后返回原始域得到所有特征值  $\lambda$ . 给定一个  $\lambda_i$ , 求解相应的特征向量相当于求解方程组  $(\lambda_i I - H)x = 0$  的非平凡解. 若  $\lambda_i$  不是重特征值, 则伴随矩阵  $(\lambda_i I - H)^* \neq O$ , 其某一行全为 0 的概率也为 0. 因此  $(\lambda_i I - H)^* e_1$  给出了一个特征向量, 且各个元素均为  $\lambda_i$  的多项式, 多项式系数由  $H$  的某些子式给出, 从而也可通过变换域计算得到. 最后, 在相差一个归一化系数的意义下,  $U$  的第  $i$  列等于  $A(\lambda_i I - H)^* e_1$ , 也为  $\lambda_i$  的多项式, 且系数由  $A$  和  $H$  的子式的乘加得到. 从而通过变换域计算可得, 再返回原始域和  $\lambda_i$  组合并归一化即可得到  $U$ .

下面以  $m = 4$  的情形具体说明. 用  $h_{i_1 i_2 \dots i_k, j_1 j_2 \dots j_k}$  表示矩阵  $H$  由  $i_1, \dots, i_k$  行、 $j_1, \dots, j_k$  列所组成的子阵的行列式. 对于 4 阶矩阵  $H = A^H A$ , 有

$$\det(\lambda I - H^H H) = \lambda^4 + c_3 \lambda^3 + c_2 \lambda^2 + c_1 \lambda + c_0,$$

其中

$$\begin{aligned} c_3 &= -\sum_{i=1}^4 h_{ii}, \\ c_2 &= \sum_{1 \leq i < j \leq 4} h_{ii}h_{jj} - h_{ij}h_{ji}, \\ c_1 &= -\sum_{1 \leq i < j < l \leq 4} h_{ijl,ijl}, \\ c_0 &= |H|. \end{aligned}$$

特征多项式的这组系数  $c$  可由变换域稀疏计算得到. 根据文献 [6, III.13] 中四次方程的解法, 特征值可经由以下步骤得出:

- (1) 求解一个三次方程  $y^3 - c_2y^2 + (c_3c_1 - 4c_0)y + (4c_2c_0 - c_3^2c_0 - c_1^2) = 0$  的根  $y$ ;
- (2) 计算

$$e = \sqrt{c_3^2/4 - c_2 + y}, \quad f = \frac{c_3y/2 - c_1}{2e};$$

- (3) 4 个特征值为

$$\begin{aligned} \lambda_{1,2} &= \frac{-c_3/2 + e \pm \sqrt{(c_3/2 - e)^2 - 2y + 4f}}{2}, \\ \lambda_{3,4} &= \frac{-c_3/2 - e \pm \sqrt{(c_3/2 + e)^2 - 2y - 4f}}{2}. \end{aligned}$$

第一步三次方程  $y^3 + Ay^2 + By + C = 0$  的一个根为 (参见文献 [6])

$$y = \sqrt[3]{-\frac{q}{2} + \frac{1}{2}\sqrt{q^2 + \frac{4p^3}{27}}} - \frac{p}{3\sqrt[3]{-\frac{q}{2} + \frac{1}{2}\sqrt{q^2 + \frac{4p^3}{27}}}} - \frac{A}{3},$$

其中

$$p = B - A^2/3, \quad q = 2A^3/27 - AB/3 + C.$$

计算出  $H$  的特征值后, 相应于每个特征值  $\lambda_i$  的特征向量 (未归一化) 为  $v_i$ , 可通过计算  $(\lambda_i I - H)^*e_1$  得到, 其每个元素均为  $\lambda_i$  的多项式, 即

$$v_i = \begin{bmatrix} \lambda_i^3 + b_{1,2}\lambda_i^2 + b_{1,1}\lambda_i + b_{1,0} \\ b_{2,2}\lambda_i^2 + b_{2,1}\lambda_i + b_{2,0} \\ b_{3,2}\lambda_i^2 + b_{3,1}\lambda_i + b_{3,0} \\ b_{4,2}\lambda_i^2 + b_{4,1}\lambda_i + b_{4,0} \end{bmatrix}.$$

通过行列式展开可以得到这组特征向量系数  $b$  为

$$\begin{aligned} b_{1,2} &= -h_{22} - h_{33} - h_{44}, & b_{1,1} &= h_{23,23} + h_{24,24} + h_{34,34}, & b_{1,0} &= -h_{234,234}, \\ b_{2,2} &= h_{21}, & b_{2,1} &= -h_{23,13} - h_{24,14}, & b_{2,0} &= h_{234,134}, \\ b_{3,2} &= h_{31}, & b_{3,1} &= -h_{34,14} + h_{23,12}, & b_{3,0} &= -h_{234,124}, \end{aligned}$$

$$b_{4,2} = h_{41}, \quad b_{4,1} = h_{34,13} + h_{24,12}, \quad b_{4,0} = h_{234,123}.$$

最后用原始矩阵  $A$  乘上  $v_i$  得到第  $i$  个左奇异向量 (未归一化), 其每个元素  $u_{ji}$  也可写成  $\lambda_i$  的多项式的形式如下:

$$u_{ji} = a_{j1}\lambda_i^3 + \sum_{t=1}^4 a_{jt}b_{t2}\lambda_i^2 + \sum_{t=1}^4 a_{jt}b_{t1}\lambda_i + \sum_{t=1}^4 a_{jt}b_{t0}.$$

因此这组左奇异向量系数也可通过变换域计算得到.

总结特征值法的过程, 我们先通过变换域稀疏计算得到特征多项式系数  $c$ 、特征向量系数  $b$  和左奇异向量系数, 再通过  $c$  求解得到原始域特征值  $\lambda_i$ ,  $1 \leq i \leq 4$ , 最后在原始域组合得到左奇异向量  $U$  并归一化得到最终的结果. 在计算特征值时, 还可以通过变换域计算得到中间量  $A, B, C$  以及  $p$  和  $q$  等, 再返回原始域组合得到结果. 还需要指出的是, 在通过行列式的展开计算  $c$  和  $b$  时, 有一部分子式的结果可以重复使用, 具体地计算方法总结在算法 7 中.

对于  $m = 2, 3$  的情形, 也存在由特征多项式直接得到特征值的求根公式, 因而也可通过与上述过程类似的方法计算出  $U$ , 以下简述这两种情形下计算  $U$  的主要步骤.

### 3.4.1 $m = 2$ 的情形

对于  $n \times 2$  阶矩阵  $A$ ,  $H = A^H A$  的特征多项式为  $\det(\lambda I - H) = \lambda^2 + c_1\lambda + c_0$ , 其中

$$c_1 = -h_{11} - h_{22}, \quad c_0 = h_{11}h_{22} - h_{12}h_{21}.$$

由此得到特征值  $\lambda_1 = \frac{-c_1 - \sqrt{c_1^2 - 4c_0}}{2}$ ,  $\lambda_2 = -c_1 - \lambda_1$ , 进而得到特征向量为

$$v_i = \begin{bmatrix} \lambda_i - h_{22} \\ h_{21} \end{bmatrix}.$$

因此左奇异向量为

$$u_{ji} = a_{j1}\lambda_i + (-a_{j1}h_{22} + a_{j2}h_{21}).$$

此时计算的具体拆分方案为: 由稀疏计算得到  $\hat{c}_1$  和  $\hat{c}_0$ ; 计算  $\lambda_{1,2}$  时, 通过稀疏计算得到  $\hat{\Delta} = \hat{c}_1^2 - 4\hat{c}_0$ , 再返回频域计算出  $\lambda_1 = \frac{-\hat{c}_1 - \sqrt{\hat{\Delta}}}{2}$ ,  $\lambda_2 = -\hat{c}_1 - \lambda_1$ ; 计算左奇异向量的方向时, 先由稀疏计算得到常数项  $\hat{d}_0 = -\hat{a}_{j1} * \hat{h}_{22} + \hat{a}_{j2} * \hat{h}_{21}$ , 再返回频域计算出  $U(j, i, :)$ , 最后在频域中对  $U$  的每列归一化.

### 3.4.2 $m = 3$ 的情形

对于  $n \times 3$  阶矩阵  $A$ ,  $H = A^H A$  的特征多项式为  $\det(\lambda I - H) = \lambda^3 + c_2\lambda^2 + c_1\lambda + c_0$ , 其中

$$c_2 = -h_{11} - h_{22} - h_{33},$$

$$c_1 = h_{12,12} + h_{13,13} + h_{23,23},$$

$$c_0 = -|H|.$$

由此得到 3 个特征值分别为

$$\lambda_1 = \sqrt[3]{-\frac{q}{2} + \frac{1}{2}\sqrt{q^2 + \frac{4p^3}{27}}} - \frac{p}{3\sqrt[3]{-\frac{q}{2} + \frac{1}{2}\sqrt{q^2 + \frac{4p^3}{27}}}} - \frac{c_2}{3},$$

**算法 7** 特征值法 ( $m = 4$  情形)输入:  $n \times 4 \times k$  三维张量  $A$ 输出:  $n \times 4 \times k$  三维张量  $U$ 

```

1:  $\hat{A} = \text{FFT}(A)$ 
2: 根据算法 1 计算  $\hat{H} = \widehat{A^H A}$ 
3:  $\hat{b}_{1,2} = -[\hat{H}(2, 2, :) + \hat{H}(3, 3, :) + \hat{H}(4, 4, :)]$ 
4:  $\hat{b}_{2,2} = \hat{H}(2, 1, :)$ ,  $\hat{b}_{3,2} = \hat{H}(3, 1, :)$ ,  $\hat{b}_{4,2} = \hat{H}(4, 1, :)$ ,  $\hat{c}_3 = \hat{b}_{1,2} - \hat{H}(1, 1, :)$ 
5: for  $1 \leq i < j \leq 4$  do
6:   根据行列式定义计算  $\hat{h}_{ij,ij}$ 
7: end for
8:  $\hat{b}_{1,1} = \sum_{2 \leq i < j \leq 4} \hat{d}_{ijij}$ ,  $\hat{c}_2 = \hat{b}_{1,1} + \sum_{j=2}^4 \hat{d}_{1j1j}$ 
9: for  $1 \leq i \leq 2, i < j \leq 4$  do
10:   根据行列式定义计算  $\hat{h}_{34,ij}$ 
11: end for
12:  $\hat{b}_{1,0} = -\hat{H}(2, 2, :) * \hat{h}_{3434} + \hat{H}(2, 3, :) * \hat{h}_{3424} - \hat{H}(2, 4, :) * \hat{h}_{3423}$ 
13:  $\hat{b}_{2,0} = \hat{H}(2, 1, :) * \hat{h}_{3434} - \hat{H}(2, 3, :) * \hat{h}_{3414} + \hat{H}(2, 4, :) * \hat{h}_{3413}$ 
14:  $\hat{b}_{3,0} = -\hat{H}(2, 1, :) * \hat{h}_{3424} + \hat{H}(2, 2, :) * \hat{h}_{3414} - \hat{H}(2, 4, :) * \hat{h}_{3412}$ 
15:  $\hat{b}_{4,0} = \hat{H}(2, 1, :) * \hat{h}_{3423} - \hat{H}(2, 2, :) * \hat{h}_{3413} + \hat{H}(2, 3, :) * \hat{h}_{3412}$ 
16: for  $j = 3 : 4$  do
17:   根据行列式定义计算  $\hat{h}_{2j,1j}$  和  $\hat{h}_{2j,12}$ 
18: end for
19:  $\hat{c}_1 = b_{1,0} - \sum_{1 < i < j \leq 4} \hat{H}(1, 1, :) * \hat{h}_{ijij} - \hat{H}(1, i, :) * \hat{h}_{ij1j} + \hat{H}(1, j, :) * \hat{h}_{ij1i}$ 
20:  $\hat{b}_{2,1} = -\hat{h}_{2414} - \hat{h}_{2313}$ ,  $\hat{b}_{3,1} = \hat{h}_{2312} - \hat{h}_{3414}$ ,  $\hat{b}_{4,1} = \hat{h}_{2412} + \hat{h}_{3413}$ 
21:  $\hat{c}_0 = -\sum_{j=1}^4 \hat{b}_{j,0} * \hat{H}(1, j, :)$ 
22:  $\hat{B} = \hat{c}_3 * \hat{c}_1 - 4\hat{c}_0$ ,  $t = \frac{1}{4}\hat{c}_3 * \hat{c}_3 - \hat{c}_2$ ,  $\hat{C} = -4\hat{t} * \hat{c}_0 - \hat{c}_1 * \hat{c}_1$ 
23:  $\hat{p} = \hat{B} - \frac{1}{3}\hat{c}_2 * \hat{c}_2$ ,  $\hat{q} = -\frac{2}{27}\hat{c}_2 * \hat{c}_2 * \hat{c}_2 + \frac{1}{3}\hat{c}_2 * \hat{B} + \hat{C}$ 
24:  $\hat{s} = \hat{q} * \hat{q} + \frac{4}{27}\hat{p} * \hat{p} * \hat{p}$ 
25:  $p = \text{IFFT}(\hat{p})$ ,  $q = \text{IFFT}(\hat{q})$ ,  $s = \text{IFFT}(\hat{s})$ ,  $c_2 = \text{IFFT}(\hat{c}_2)$ 
26:  $u = \sqrt[3]{(-q + \sqrt{s})/2}$ 
27:  $y = u - \frac{p}{3u} + \frac{c_2}{3}$ 
28:  $t = \text{IFFT}(t)$ ,  $c_3 = \text{IFFT}(c_3)$ ,  $c_1 = \text{IFFT}(c_1)$ 
29:  $e = \sqrt{t + y}$ ,  $f = \frac{c_3 \circ y - 2c_1}{4e}$ 
30:  $\lambda_1 = \frac{-c_3/2 + e + \sqrt{(c_3/2 - e)^2 - 2y + 4f}}{2}$ ,  $\lambda_3 = \frac{-c_3/2 - e + \sqrt{(c_3/2 + e)^2 - 2y - 4f}}{2}$ 
31:  $\lambda_2 = -c_3/2 + e - \lambda_1$ ,  $\lambda_4 = -c_3/2 - e - \lambda_3$ 
32: for  $j = 1 : n$  do
33:    $d_3 = \text{IFFT}(\hat{A}(j, 1, :))$ 
34:   for  $i = 1 : 3$  do
35:      $\hat{d}_j = \sum_{t=1}^4 \hat{A}(j, t, :) * b_{t,i}$ 
36:      $d_i = \text{IFFT}(\hat{d}_i)$ 
37:   end for
38:   for  $i = 1 : 4$  do
39:      $U(j, i, :) = d_3 \circ \lambda_i^3 + d_2 \circ \lambda_i^2 + d_1 \circ \lambda_i + d_0$ 
40:   end for
41: end for
42: normalize  $U$ 

```

$$\lambda_2 = e^{2\pi i/3} \sqrt[3]{-\frac{q}{2} + \frac{1}{2}\sqrt{q \circ q + \frac{4p \circ p \circ p}{27}}} - e^{4\pi i/3} \frac{p}{3 \sqrt[3]{-\frac{q}{2} + \frac{1}{2}\sqrt{q^2 + \frac{4p^3}{27}}}} - \frac{c_2}{3},$$

$$\lambda_3 = e^{4\pi i/3} \sqrt[3]{-\frac{q}{2} + \frac{1}{2}\sqrt{q^2 + \frac{4p^3}{27}}} - e^{2\pi i} \frac{p}{3\sqrt[3]{-\frac{q}{2} + \frac{1}{2}\sqrt{q^2 + \frac{4p^3}{27}}}} - \frac{c_2}{3},$$

其中  $p = c_1 - c_2^2/3$ ,  $q = 2c_1^2/27 - c_1c_2/3 + c_0$ . 进而得到特征向量为

$$v_i = \begin{bmatrix} \lambda_i^2 + (-h_{22} - h_{33})\lambda_i + h_{23,23} \\ h_{21}\lambda_i - h_{23,13} \\ h_{31}\lambda_i + h_{23,12} \end{bmatrix},$$

因此左奇异向量为

$$u_{ji} = a_{j1}\lambda_i^2 + [-a_{j2}(h_{22} + h_{33}) + a_{j2}h_{21} + a_{j3}h_{31}]\lambda_i + a_{j1}h_{23,23} - a_{j2}h_{23,13} + a_{j3}h_{23,12}.$$

此时计算的具体拆分方案为: 由稀疏计算得到  $\hat{c}_0$ ,  $\hat{c}_1$  和  $\hat{c}_2$ ; 计算 3 个特征值时, 先由稀疏计算得到  $\hat{p}$  和  $\hat{q}$ , 再由此得到根号内的部分  $\hat{s} = \hat{q} * \hat{q} + \frac{4}{27}\hat{p} * \hat{p}$ , 然后返回频域计算出

$$u = \sqrt[3]{(-q + \sqrt{s})/2}, \quad v = -p/3u,$$

最后得到 3 个特征值为

$$\lambda_1 = u + v - c_2/3, \quad \lambda_2 = e^{2\pi i/3}u + e^{4\pi i/3}v - c_2/3, \quad \lambda_3 = e^{4\pi i/3}u + e^{2\pi i/3}v - c_2/3.$$

计算奇异向量的方向时, 先由稀疏计算得到  $U(j, i, :)$  作为  $\lambda_i$  的多项式的系数, 再返回频域得到  $U(i, j, :)$ , 最后在频域中对  $U$  的每列归一化.

对于  $m \geq 5$  的情形, 首先奇异值的计算将涉及一个超过 5 次的多项式求根, 需使用迭代的方法, 另外通过行列式展开计算特征多项式和特征向量系数的复杂度将迅速提升, 也需要寻找其他计算方法.

## 4 直接补零框架的矩阵稀疏运算

### 4.1 直接补零框架

前文讨论了矩阵稀疏计算的基本框架, 其中只包含加减乘以及复共轭的部分可在变换域中直接稀疏计算得到, 但需按照具体的展开形式进行计算 (例如, 行列式按照组合定义计算, 特征多项式由子式相加得到), 因此对于  $n$  阶矩阵, 其复杂度将超过  $n!$  量级, 当  $n$  较大时很可能不如传统计算. 一种解决方式是, 对传统计算进行拆分, 例如利用算法 3 计算行列式, 但这一思路不易推广至一般情形.

事实上, 我们还有如下不依赖于具体的乘加展开形式的变换域计算框架:

**定理 4.1** 设  $f(A)$  为只包含矩阵元素加减乘的连续矩阵簇运算, 且  $f$  作为矩阵元素的多元多项式是齐次的,  $\widehat{f(A)}$  的非零元个数  $p$  不超过矩阵簇的长度, 则  $\widehat{f(A)}_{nz} = \mathcal{F}[f(\mathcal{F}^{-1}(\hat{A}_{nz+p}))]$ , 其中  $\hat{A}_{nz+p}$  表示  $\hat{A}$  的非零部分补零所得到的矩阵簇.

为了说明上述流程的正确性, 回顾此前向量乘法的变换域计算所得的结论

$$\widehat{a \circ b}_{nz+p'} = \mathcal{F}[\mathcal{F}^{-1}(\hat{a}_{nz+p'}) \circ \mathcal{F}^{-1}(\hat{b}_{nz+p'})],$$

其中  $p' \geq p_a + p_b - 1$ . 因此, 对于 3 个向量的乘积  $a \circ b \circ c$ , 如果其 Fourier 系数的非零元长度不超过  $p$ , 则  $\widehat{a \circ b}$  的稠密度也不超过  $p$ , 从而对于任意的  $p' \geq p$ , 成立

$$\begin{aligned} (\mathcal{F}(a \circ b \circ c))_{nz+p'} &= \mathcal{F}[\mathcal{F}^{-1}(\widehat{a \circ b}_{nz+p'}) \circ \mathcal{F}^{-1}(\hat{c}_{nz+p'})] \\ &= \mathcal{F}[\mathcal{F}^{-1}(\hat{a}_{nz+p'}) \circ \mathcal{F}^{-1}(\hat{b}_{nz+p'}) \circ \mathcal{F}^{-1}(\hat{c}_{nz+p'})]. \end{aligned}$$

类似地可以归纳证明, 对于  $n$  个向量的点乘  $a^1 \circ \dots \circ a^n$ , 如果其结果的 Fourier 系数的稠密度不超过  $p$ , 则对任意的  $p' \geq p$ , 成立

$$(\mathcal{F}(a^1 \circ \dots \circ a^n))_{nz+p'} = \mathcal{F}[\mathcal{F}^{-1}(\hat{a}_{nz+p'}^1) \circ \dots \circ \mathcal{F}^{-1}(\hat{a}_{nz+p'}^n)].$$

由 Fourier 变换的线性性质即可得到, 对于一般只包含加减乘运算的  $f$ , 定理结论成立.

对于只包含加减乘运算的函数  $f$ , 与第 2.3 小节中的等价计算框架相比, 定理 4.1 的直接补零框架使得我们只需要对  $\mathcal{F}^{-1}(A_{nz+p})$  中的每个矩阵计算  $f$  即可, 因此可以自由地使用除法和开方等运算, 一个最典型的例子是行列式的计算, 直接补零框架使用传统的 LU 分解即可.

但另一方面, 直接补零框架必须直接补足结果的稠密度, 因此结果较为稠密时这一方法可能失效, 而等价计算中稠密度逐渐增长, 可在一开始稠密度较低时稀疏计算, 而在稠密度较高时切换到原始域传统计算. 因此, 需要权衡使用或综合使用两类框架.

最后, 定理 4.1 中的运算  $f$  如果包含矩阵元素的复共轭运算, 则定理不成立, 但一般对于有变换域稀疏性的向量  $a = \bar{b}$ , 记其稠密度为  $p$ , 将  $\hat{a}$  和  $\hat{b}$  的非零部分均补零至  $p'$  并进行 IFFT, 得到的向量  $\tilde{a}$  和  $\tilde{b}$  满足

$$\tilde{a}_i = s_i \tilde{b}_i,$$

其中

$$s_i = e^{2\pi\sqrt{-1}i(p-1)/p'}.$$

根据这一点可快速地得到  $\tilde{b}$ , 从而通过将  $\tilde{a}$  视为另一组向量从而可以使用定理. 由于这里的常数  $s_i$  只与位置有关而与  $a$  和  $b$  的具体取值无关, 因此若矩阵簇  $A$  和  $B$  满足  $A = \bar{B}$ , 则成立  $\tilde{A}_i = s_i \tilde{B}_i$ , 特别地, 对于 Hermite 矩阵,  $s_i^{-1/2} \tilde{A}_i$  也是 Hermite 矩阵.

以下考虑将直接补零框架应用于矩阵 SVD 计算.

## 4.2 对称特征值问题与一般矩阵 SVD

我们将特征值法应用到一般情形, 在特征值法中, SVD 计算分为矩阵乘法  $H = A^H A$ 、对称特征值分解  $H = V \Sigma^2 V^H$  和拆分的矩阵乘法  $U = AV$ . 首先考虑对称特征值问题的求解.

一般地, 对于一个  $m$  阶矩阵  $H$ , 其特征多项式  $p(\lambda) = |\lambda I - H| = \lambda^m + c_{m-1} \lambda_{m-1} + \dots + c_0$ , 可以使用多项式插值的方法计算系数, 即采样  $m$  个点  $\sigma_1, \dots, \sigma_m$ , 计算  $p(\sigma_i) = |\sigma_i I - H|$ , 再通过求解如下线性方程组得到系数  $c$ :

$$\begin{bmatrix} 1 & \sigma_1 & \cdots & \sigma_1^{m-1} \\ 1 & \sigma_2 & \cdots & \sigma_2^{m-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \sigma_m & \cdots & \sigma_m^{m-1} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{m-1} \end{bmatrix} = \begin{bmatrix} p(\sigma_1) - \sigma_1^m \\ p(\sigma_2) - \sigma_2^m \\ \vdots \\ p(\sigma_m) - \sigma_m^m \end{bmatrix}.$$

使用采样法计算特征多项式有一些需要注意的细节. 首先对于  $\sigma_i$  的选取, 由于上述线性方程组的系数矩阵是 Vandermonde 矩阵, 根据文献 [7] 中的结论, 为了控制其条件数, 只有选取  $\sigma_i = e^{-2\pi\sqrt{-1}i/m}$ , 此时系数矩阵正是 FFT 矩阵, 条件数为 1. 另外, 即使系数矩阵为正交矩阵, 但得到的特征多项式系数仍可能不精确, 例如,  $H$  的特征值都比较接近 0 的情形, 此时  $c_{m-1}$  的数量级远小于  $p(\sigma_i) - 1$  的数量级, 由于有效数字相消,  $c_{m-1}$  的相对精度将较低, 以至于多项式求根时得到错误的结果. 一个可行的解决方法是对  $H$  进行约化以使其特征值接近于 1, 例如, 对  $H/\|H\|_F$ , 计算其特征多项式系数  $\tilde{c}$ , 再由  $c_i = \tilde{c}_i\|H\|_F^{m-i}$  得到原矩阵的特征多项式系数.

如果应用等价计算框架, 对于一个  $m \times m \times k$  的张量  $H$ , 对各个位置的矩阵选取相同的采样点计算特征多项式系数, 则  $\sigma I - H$  的 Fourier 系数也是稀疏的, 从而可以通过变换域的 LU 分解得到行列式  $|\sigma_i I - H|$  并求解得到特征多项式系数. 但变换域的 LU 分解的复杂度为  $O(pm^4 \log(mp))$ , 共有  $m$  个行列式需要计算, 因此总的复杂度为  $O(pm^5 \log(mp))$ . 而直接计算行列式的复杂度为  $O(km^3)$ , 因此需要  $m^2 p \log(mp)$  相对于  $k$  较小才有加速, 实际的例子的稀疏性一般很难满足, 必须寻找更低复杂度的算法.

由于特征多项式系数  $c$  可表达为主子式之和, 只涉及矩阵元素的加减乘运算, 因此可以使用直接补零的框架进行计算.  $\hat{c}$  中最大的稠密度为  $mp$ , 回顾定理 4.1, 只需要对  $\mathcal{F}^{-1}(\hat{H}_{nz+})$  中的矩阵  $\tilde{H}$  (共  $mp$  个) 计算其特征多项式的系数, 再计算 FFT 即可得到  $\hat{c}$ . 对于单个矩阵的特征多项式系数计算, 仍采用基于多项式插值的方法, 但不再单独计算  $\sigma_i I - \tilde{H}$ , 而是先对  $\tilde{H}$  进行上 Hessenberg 化, 得到  $\tilde{H} = QTQ^H$ . 事实上, 根据 Fourier 系数与共轭的关系, 当  $\tilde{H}$  为 Hermite 阵时,  $T$  是 (非 Hermite 的) 三对角矩阵. 在三对角化后, 由于

$$|\sigma_i I - \tilde{H}| = |\sigma_i I - QTQ^H| = |\sigma_i I - T|,$$

因此只需要对  $m$  个三对角矩阵计算行列式, 并反解得到系数即可. 对于单个矩阵, 这一过程总共需要  $m^3$  的运算量, 因此计算  $\hat{c}$  的复杂度为  $O(pm^4 \log(mp))$ , 降低了一个数量级.

对于特征向量系数, 我们类似地通过多项式插值的方式计算. 给定特征值  $\lambda_i$ , 其对应的特征向量由  $(\lambda_i I - H)^* e_1$  给出, 可写作  $\lambda_i$  的多项式的形式如下:

$$v_i = \begin{bmatrix} \lambda_i^{m-1} + b_{1,m-2}\lambda_i^{m-2} + \cdots + b_{1,0} \\ b_{2,m-2}\lambda_i^{m-2} + \cdots + b_{2,0} \\ \vdots \\ b_{m,m-2}\lambda_i^{m-2} + \cdots + b_{m,0} \end{bmatrix},$$

其中  $v_i$  的每个位置上都有  $m-1$  个待定系数. 因此只需要在  $m-1$  个点采样计算即可解出, 我们仍然选择所有的  $m-1$  次单位根作为采样点, 系数  $b$  可通过求解以下线性方程组得到:

$$\begin{bmatrix} 1 & \sigma_1 & \cdots & \sigma_1^{m-2} \\ 1 & \sigma_2 & \cdots & \sigma_2^{m-2} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \sigma_m & \cdots & \sigma_m^{m-2} \end{bmatrix} \begin{bmatrix} b_{1,0} & \cdots & b_{m,0} \\ b_{1,1} & \cdots & b_{m,1} \\ \vdots & \ddots & \vdots \\ b_{1,m-2} & \cdots & b_{m,m-2} \end{bmatrix} = [(\sigma_1 I - H)^* e_1 \cdots (\sigma_m I - H)^* e_1]^T - \mathbf{1} e_1^T,$$

其中  $\mathbf{1}$  表示全为 1 的列向量.

同样地, 我们用直接补零的框架计算特征向量系数  $b$ , 对于  $\mathcal{F}^{-1}(H_{nz+})$  中的每个矩阵  $\tilde{H}$  计算  $b$ , 再由 FFT 得到  $\hat{b}$ . 为了计算  $(\sigma_i I - \tilde{H})^* e_1$ ,  $1 \leq i \leq m-1$ , 先进行三对角化  $\tilde{H} = QTQ^H$ , 并根据

$$(\sigma_i I - \tilde{H})^* e_1 = |\sigma_i I - \tilde{T}|(\sigma_i I - H)^{-1} e_1 = |\sigma_i I - \tilde{T}|Q(\sigma_i I - T)^{-1}Q^H e_1$$

计算, 由于三对角矩阵的行列式和线性方程组求解只需要  $O(m)$  的运算量, 因此计算所有  $(\sigma_i I - \tilde{H})^* e_1$  的运算量, 并求解出特征向量系数的运算量为  $O(m^3)$ , 求解出  $\hat{b}$  的复杂度为  $O(pm^4 \log(mp))$ . 另外, 尽管系数  $b$  中最大的稠密度为  $(m-1)p$ , 但为了重复利用先前计算特征多项式的三对角化结果, 可以仍选择将  $\hat{H}_{nz}$  补零至长度为  $mp$ .

对于特征值的计算, 我们采用 Newton 迭代法, 一般而言, 使用 Newton 迭代法求多项式的根无法保证得到所有的解, 但在  $H$  为 Hermite 阵的条件下, 其特征值均为实数, 根据以下基本事实可求得最小的根.

**基本事实 4.1**  $p(x) = (x - \lambda_1)(x - \lambda_2) \cdots (x - \lambda_m)$  为多项式,  $\lambda_1 < \lambda_2 < \cdots < \lambda_n$ , 设  $x_0 < \lambda_1$ , 则 Newton 迭代

$$x_{k+1} = x_k - \frac{p(x_k)}{p'(x_k)}$$

收敛于  $\lambda_1$ .

在求得最小的根后, 通过多项式的除法  $p(x)/(x - \lambda_1)$  将得到的因子约去, 则新的多项式最小的根为  $\lambda_2$ , 因此从  $\lambda_1$  开始进行迭代将收敛于  $\lambda_2$ , 依次进行可求得全部的根. 由于 Newton 迭代涉及较多的除法且难以约分, 因此可在原始域中逐个矩阵计算, 得到其特征值, 最后再在原始域中代入特征值得到特征向量  $V$  并归一化, 完成  $H$  的特征值分解.

对于矩阵 SVD 问题, 首先与低阶 SVD 一样通过变换域的矩阵乘法得到  $\hat{H} = \widehat{A^H A}$ , 然后通过上述推广的特征值法得到特征多项式系数  $\hat{c}$  和特征向量系数  $\hat{b}$  并在原始域上得到特征值  $\lambda$ . 对于左奇异向量  $U$  的计算, 其元素可表示为

$$u_{ji} = a_{ji} \lambda_i^{m-1} + \sum_{t=0}^{m-2} d_{jt} \lambda_i^t,$$

其中

$$d_{jt} = \sum_{s=1}^m a_{js} b_{st}$$

可通过变换域稀疏计算得到, 因此通过  $\hat{A}$  和  $\hat{b}$  计算得到  $\hat{d}$ , 再返回原始域和  $\lambda$  组合并归一化即可得到左奇异向量  $U$ .

### 4.3 其他问题的直接补零方法

此前介绍了 SVD 问题使用直接补零框架的算法, 事实上此前介绍的变换域 LU 分解等问题也可通过直接补零的框架得到变换域算法, 由于其形式比较简单, 拆分形式在之前的有关章节中也已得到, 本小节对这些算法一并说明.

对于  $n$  阶矩阵构成的矩阵簇的 LU 分解, 其拆分形式为

$$A = LU = (L_f D_L^{-1})(D_U^{-1} U_f),$$

其中  $L_f$ ,  $U_f$ ,  $D_U$  和  $D_L$  都由矩阵  $A$  的元素通过加减乘运算得到,  $D_L$  和  $D_U$  为对角阵, 其第  $i$  个对角元分别为  $d_i$  和  $d_{i-1}$ ,  $d_i$  表示  $A$  的前  $i$  行前  $i$  列构成的子矩阵的行列式. 因此, 在直接补零变换域 LU

算法中, 将  $\hat{A}$  补零至  $np$  并计算  $\mathcal{F}^{-1}(\hat{A}_{nz+})$ , 对其中的每一个矩阵计算其传统的 LU 分解  $\tilde{A} = LU$ , 因为  $U$  的对角元为  $u_{ii} = d_i/d_{i-1}$ , 由此得到  $d_i = \prod_{j=1}^i u_{jj}$ , 即得到了  $\tilde{D}_L$  和  $\tilde{D}_U$ , 进而得到

$$\tilde{L}_f = LD_L, \quad \tilde{U}_f = D_U U.$$

最后将计算得到的张量  $\tilde{L}_f, \tilde{U}_f$  和  $\tilde{D}_L$  进行一次 IFFT 即可得到与算法 3 一致的结果.

对于 Cholesky 分解, 尽管 Hermite 矩阵簇  $A$  经 FFT、截断、补零、再 IFFT 后得到的矩阵簇不再是 Hermite 的, 但存在常数  $s_i$  使得  $(s_i \tilde{A}_i)$  为 Hermite 阵, 通过对这个 Hermite 矩阵进行 LDLT 分解容易得到  $\tilde{A}_i$  的 LU 分解和顺序主子式的值, 进而得到  $\tilde{L}_f$  和  $\tilde{D}_L$ .

对 LU 分解得到的  $L$  求逆, 其拆分形式为

$$L^{-1} = D_U^{-1} K_f.$$

因此对于  $\mathcal{F}^{-1}(A_{nz+})$  中的每个矩阵, 在 LU 分解的结果的基础上计算得到  $\tilde{K}_f = D_U L^{-1}$ , 再进行 IFFT 即可.

变换域的矩阵求逆算法也可类似地得到, 其拆分形式为

$$A^{-1} = K_f / |A|.$$

对  $\mathcal{F}^{-1}(A_{nz+})$  中的每个矩阵基于 LU 分解求得行列式  $|\tilde{A}|$  和  $\tilde{A}^{-1}$ , 由此得到  $\tilde{K}_f = |\tilde{A}| \tilde{A}^{-1}$ , 并对  $\tilde{K}_f$  和  $|\tilde{A}|$  进行 IFFT 即得到结果.

最后, 对于 QR 分解, 其拆分形式为

$$A = QR = (AK_f(D_L D_U)^{-1/2})((D_L D_U)^{-1/2} L_f^H).$$

因此, 对于直接补零的框架, 先从  $\mathcal{F}^{-1}(\hat{A}_{nz+})$  得到  $\mathcal{F}^{-1}(\hat{A}_{nz+}^H)$ , 再综合利用 Cholesky 分解和矩阵求逆的方法得到  $\mathcal{F}^{-1}(\widehat{AK}_{f_{nz+}})$ ,  $\tilde{D}_L$  和  $\tilde{L}_f$  再反变换即可.

## 5 矩阵运算的稀疏近似计算

在前面的章节中, 变换域的稀疏算法对于截断之后的张量在不考虑数值计算的误差下是精确的, 即对于张量  $A$ , 如果其 Fourier 系数只有  $p$  个位置非零, 则变换域计算得到精确的  $f(A)$ , 而对于一般的张量, 只有将  $\hat{A}$  范数较小的部分截去所产生的误差. 这类精确的方法可以较好地估计误差, 但事实上在很多例子中, 矩阵运算的结果往往也具有 Fourier 变换下的稀疏性, 因此精确方法中乘法所造成的稠密度增加, 拆分方法中频域的组合等运算可能是冗余的, 而一些近似的方法可以在精度相近的情形下更高效, 本节简单讨论一些近似计算的可能方法.

### 5.1 采样算法的 Fourier 插值

一种最直接利用矩阵运算结果的 Fourier 变换稀疏性的方法是利用采样计算和插值. 为了说明采样插值的方法, 先考虑一个向量  $x$ , 其满足变换域的稀疏性假设, 则有

$$x_q = \sum_{j=-l_x}^{s_x-1} \hat{x}_j e^{2\pi\sqrt{-1}jq/k},$$

其中  $l_x$  和  $s_x$  分别表示  $\hat{x}$  的首末非零元的个数,  $p_x = l_x + s_x$  为  $\hat{x}$  的稠密度. 现假设我们已经对于某个包含  $p_x$  个元素的指标集  $\mathcal{I}$  中的所有  $q$  得到了  $x_q$ , 则根据上式可以得到一个关于  $\hat{x}$  的  $p_x$  阶线性方程组, 进而可以解出 Fourier 系数的非零部分.

因此, 对于矩阵簇计算问题  $f(A)$ , 如果假设  $\widehat{f(A)}$  具有变换域稀疏性, 稠密度为  $p$ , 且非零元的位置也已确定, 则可以通过对一个包含  $p$  个元素的指标集  $\mathcal{I}$  计算, 再通过求解线性方程组得到  $\widehat{f(A)}$ . 这一过程需要对  $p$  个矩阵进行常规的矩阵计算并求解线性方程组, 由于对固定的指标集和固定的非零元位置, 线性方程组的系数矩阵也是固定的, 因此线性方程组求解本质上是矩阵向量乘法, 但仍需要  $O(p^2)$  的运算量.

考虑通过适当的采样降低线性方程组的运算量, 假设  $\widehat{f(A)}$  的稠密度  $p$  整除  $k$ , 记  $d = k/p$ , 选取采样点  $\mathcal{I} = \{0, d, \dots, (p-1)d\}$ , 则此时线性方程组形如

$$[x_0 \ x_d \ \cdots \ x_{(p-1)d}]^T = F[\hat{x}_0 \ \cdots \ \hat{x}_{s_x-1} \ \hat{x}_{-l_x} \ \cdots \ \hat{x}_{-1}]^T,$$

其中系数矩阵  $F$  的元素为

$$\begin{aligned} f_{ij} &= e^{2\pi\sqrt{-1}idj/k} = e^{2\pi\sqrt{-1}ij/p}, \quad 0 \leq j \leq s_x - 1, \\ f_{ij} &= e^{2\pi\sqrt{-1}id(j-p)/k} = e^{2\pi\sqrt{-1}ij/p}, \quad s_x \leq j \leq p_x - 1, \end{aligned}$$

即  $F$  恰为  $p$  阶的 IFFT 矩阵. 因此在均匀采样下, 通过插值得到  $\hat{x}_{nz}$  的非零部分也可通过 FFT 实现.

作为一个例子, 考虑 Fourier 插值的 SVD 算法, 回顾矩阵  $A$  的左奇异向量可由  $U = AV\Sigma^{-1}$  得到, 因此可以通过 Fourier 插值的方法得到  $\widehat{V\Sigma^{-1}}$ , 再通过变换域矩阵乘法得到  $\hat{U}$ . 需要指出的是, 对采样得到的  $(V\Sigma^{-1})_{\mathcal{I}_q}$ ,  $0 \leq q \leq p-1$  直接进行 FFT 得到 Fourier 系数的非零部分往往误差较大, 需要对相邻矩阵的同一个特征向量进行相位对齐, 即对  $(V\Sigma^{-1})_{\mathcal{I}_{q+1}}$  的第  $j$  列乘以一个模长为 1 的数, 使得其与  $(V\Sigma^{-1})_{\mathcal{I}_q}$  的第  $j$  列的内积为实数. 加入相位对齐后, 在很多测试数据集上有较好的精度.

基于采样的 Fourier 插值法运算量较低, 主要的计算只在于对  $p$  个矩阵进行常规的矩阵运算, 但由于算法依赖于结果的稀疏性, 而结果的稀疏性往往只有经验的结果, 一般难以较好地事先估计, 因此得到的近似结果正确性无法保证, 对于 SVD 问题的 Fourier 插值法也在部分数据集上出现了较大的误差.

## 5.2 迭代算法

一些矩阵运算还可以通过迭代法的方式求解, 以特征值分解为例, 可将特征对  $(\lambda, x)$  的计算转化为二次方程组的求解, 进而用 Newton 法求解方程组得到特征对.

以计算特征向量的一类二次方法为例 (参见文献 [5]), 考虑  $z$  为一个固定向量,  $(\lambda, x)$  为矩阵  $A$  的一个特征对, 且  $z^H x = C$  为非零常数, 则有

$$\lambda = \langle z, Ax \rangle / C = \langle A^H z, x \rangle / C.$$

记  $w = A^H z$ , 则特征向量  $x$  为以下向量值函数的零点:

$$F(x) = Ax - \frac{\langle w, x \rangle}{C} x.$$

用 Newton 迭代法求解这一方程组, 并每一步迭代中保证  $\langle z, x \rangle = C$ , 则得到迭代格式为

$$\lambda_l = \frac{\langle w, x_l \rangle}{C},$$

$$\begin{aligned}
 J_l &= A - \lambda_l I - \frac{x_l w^H}{C}, \\
 \tilde{x}_{l+1} &= x_l - J_l^{-1} F(x_l), \\
 x_{l+1} &= \frac{C \tilde{x}_{l+1}}{\langle z, \tilde{x}_{l+1} \rangle}.
 \end{aligned}$$

由于迭代只涉及确定的求逆和除法运算, 因此可以拆分为分子、分母两部分进行计算, 分别记为  $x_l^{\text{nu}}$  和  $x_l^{\text{de}}$ , 则相应的迭代格式为

$$\begin{aligned}
 J_l^{\text{nu}} &= C x_l^{\text{de}} A - \langle w, x_l^{\text{nu}} \rangle I - x_l^{\text{nu}} w^H, \\
 \frac{K_l}{d_l} &= (J_l^{\text{nu}})^{-1}, \\
 \tilde{x}_{l+1}^{\text{de}} &= d_l x_l^{\text{de}}, \\
 \tilde{x}_{l+1}^{\text{nu}} &= d_l x_l^{\text{nu}} - C x_l^{\text{de}} K_l A x_l^{\text{nu}} + K_l \langle w, x_l^{\text{nu}} \rangle x_l^{\text{nu}}, \\
 x_{l+1}^{\text{nu}} &= C \tilde{x}_{l+1}^{\text{de}} \tilde{x}_{l+1}^{\text{nu}}, \\
 x_{l+1}^{\text{de}} &= \tilde{x}_{l+1}^{\text{de}} * \langle z, \tilde{x}_{l+1}^{\text{nu}} \rangle,
 \end{aligned}$$

其中的分数线表示拆分, 而非计算除法.

我们希望通过较好的初值 (如前述的 Fourier 插值法的结果) 作为初值, 在较少的迭代步内得到特征分解, 但目前还有一些问题有待解决. 首先, 对上述二次方法, 即使初值  $x$  已经与某个特征向量有良好的相关性, 仍可能收敛于其他的特征向量, 这导致迭代法难以确保求得所有特征对, 一些可能的解决方案参见文献 [1], 但变换域的方案还需进一步探索; 另外, 在迭代过程中稠密度迅速增加, 精确计算迭代一步的稠密度将增加  $n$  倍,  $n$  为矩阵阶数. 为此, 一种可能的方法在中间步骤适当地进行截断, 但具体实现与效果也有待进一步研究.

## 6 数值实验

本节对华为无线网络 RAN 算法部提供的数据在 MATLAB 环境下对部分算法进行数值实验, 以展示变换域计算的精度和相比直接计算的方法的效率, 用于计算的张量  $A$  维数为  $64 \times 4 \times 816$ , 其变换域结果  $\hat{A}$  中各位置矩阵的 Frobenius 范数如图 5 所示.

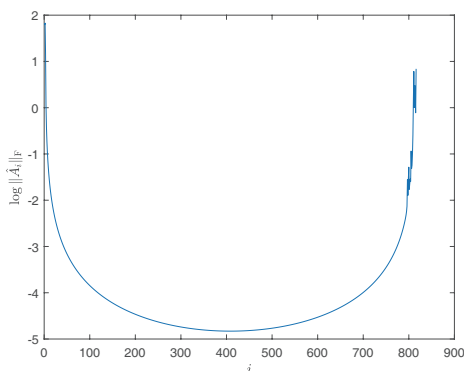


图 5 (网络版彩图)  $\hat{A}$  稀疏性

首先对于矩阵乘法的变换域算法进行实验, 比较直接计算  $B = A^H A$  及其变换域计算. 取连续的非零元位置为前后各  $p/2$  个, 并对不同的  $p$  比较其与直接计算之间的时间 (以循环 1,000 次所需时间之比表示), 并计算平均的相对误差 (以  $\|B_{\text{approx}} - B\|_F / \|B\|_F$  表示), 相关结果如表 1 所示.

其次是矩阵求逆  $B^{-1}$  的计算, 采取直接补零框架的变换域计算, 相关结果如表 2 所示.

对于 QR 分解, 分别对每个矩阵单独求解得到正交阵  $Q$ , 由直接补零框架变换域计算得到  $Q_{\text{approx}}$ , 将所有夹角的余弦值  $Q(:, i, j)^T Q_{\text{approx}}(:, i, j)$ ,  $i = 1, 2, \dots, m$ ,  $j = 1, 2, \dots, k$  的平均值作为精度, 相关结果如表 3 所示. 在结果稠密度不太大时, LU (Cholesky) 分解、三角矩阵求逆的变换域算法也有类似的加速效果.

最后, 对于 SVD 问题, 我们同样考虑左奇异向量与精确结果的夹角余弦值, 以去除 5% 的最低精度后, 剩余位置的最低精度作为标准. 显然, 精度越接近 1 代表计算结果越准确.

我们先考虑特征值法, 在变换域中取  $p = 16$  进行实验, 由  $\hat{H}$  计算得到  $U$ , 由于在当前设定下变换域计算左奇异向量系数 (算法 7 中的  $\hat{d}_i$ ) 没有速度上的收益, 因此直接由系数  $\hat{b}_{ij}$  返回原始域后计算得到  $V$ , 进而由矩阵乘法得到  $U$ . 将其与逐个计算的方法对比用时 (由于  $m$  远小于  $n$ , 因此统计了反变换  $O(m^2 k \log k)$  复杂度对应的时间), 结果如表 4 所示.

我们也对于 Fourier 域插值法进行测试, 均匀采样 34 个位置的矩阵进行计算得到  $\widehat{V\Sigma^{-1}}$ , 再与截断到  $p = 30$  的  $\hat{H}$  进行变换域矩阵乘得到  $\hat{U}$ , 实验结果如表 4 所示. 另外, 算法中没有对结果  $U$  进行归一化操作, 但相应导致的模长与 1 之间的平均误差也仅为

$$\sum \| \|U(:, i, j)\| - 1 \| / (4 \times 816) = 0.0241.$$

表 1 变换域矩阵协方差测试

$p$	20	30	40	60	80
误差	5.67%	3.72%	2.88%	1.88%	1.49%
加速倍数	3.25	2.88	2.47	1.98	1.66

表 2 变换域矩阵求逆测试

$p$	20	30	40	60	80
误差	12.59%	5.27%	3.02%	1.86%	1.55%
加速倍数	4.30	2.84	1.89	1.30	0.61

表 3 变换域 QR 分解测试

$p$	10	20	30	40
精度	0.9458	0.9854	0.9909	0.9955
加速倍数	2.80	2.09	1.50	0.92

表 4 变换域 SVD 测试

方法	第一个奇异向量	第二个奇异向量	第三个奇异向量	第四个奇异向量	加速倍数
特征值法	0.9988	0.9801	0.9726	0.9445	1.29
Fourier 域插值法	0.9949	0.9656	0.9659	0.9661	4.03

表 5 部分矩阵运算的稀疏计算算法概况

运算类型	表达式	稀疏计算	特殊处理	稠密度 (上界)
矩阵乘积	$Ax, AB$	Y	N	$2p$
协方差	$AA^H$	Y	N	$2p$
行列式	$\det(A)$	Y	N	$np$
矩阵逆	$A^{-1}$	Y	Y	$np$
投影算子	$A(A^H A)^{-1} A^H$	Y	Y	$(n+2)p$
Cholesky 分解	$A = LL^T$	Y	Y	$np$
SVD	$A = U\Sigma V^H$	Y	Y	$2np$

在无线通信的实际应用中, SVD 的精度有大部分在 0.9 或 0.95 以上可基本满足应用要求, 因此上述两种变换域 SVD 算法在本例实验数据上都符合这一标准.

## 7 结论及展望

在原始数据具有时域稀疏性的假设下, 本文从理论上证明了矩阵操作的可稀疏计算, 从而可在时域高效实现, 并给出了几类部分矩阵运算的算法方案, 如表 5 所示, 对于其中的基本运算, 通过数值实验验证了理论和算法的正确性.

同时也注意到对于现有变换域算法, 在精度方面, 部分复杂拆分操作可能不够细致, 算法的精度与稳定性可能受到影响; 在复杂度方面, SVD 等复杂的操作需要进行拆分与原始域组合, 稠密度也随着乘法不断增加, 这些特点使相关变换域算法的复杂度难以进一步降低, 但事实上对于实际数据, 由于存在连续性且变换较慢, 其 Fourier 系数在稀疏性之外还进一步具备衰减性等特点, 因此多次乘法后的结果, 甚至除法或开方运算的结果仍具备稀疏性, 利用上述性质可能得到更稀疏的近似结果和更精简的拆分形式, 进而提高算法速度, 其中一类初步结果已在 Fourier 域插值法中得到验证.

针对以上提出的问题, 后续的研究可以包括下列方向:

- 各类复杂算子拆分方案的数值稳定性分析和改进;
- 稀疏域迭代类算法设计;
- 运算中间量和结果的稀疏性利用,

进而完善和发展现有算法方案, 进一步提高算法性能和精度.

## 参考文献

- 1 Baik R, Datta K, Hong Y. Group homotopy algorithm with a parameterized Newton iteration for symmetric eigen problems. In: Computational Science. Lecture Notes in Computer Science, vol. 351. Berlin-Heidelberg: Springer, 2005, 899–907
- 2 Demmel J. Applied Numerical Linear Algebra. Philadelphia: SIAM, 1997
- 3 Kernfeld E, Kilmer M, Aeron S. Tensor-tensor products with invertible linear transforms. *Linear Algebra Appl*, 2015, 485: 545–570
- 4 Kilmer M E, Martin C D. Factorization strategies for third-order tensors. *Linear Algebra Appl*, 2011, 435: 641–658
- 5 Kondo K, Yasukouchi S, Iwasaki M. Eigendecomposition algorithms solving sequentially quadratic systems by Newton method. *JSIAM Lett*, 2009, 1: 40–43
- 6 Morandi P. Field and Galois Theory. New York: Springer, 1996

7 Pan V Y. How bad are Vandermonde matrices? [SIAM J Matrix Anal Appl](#), 2016, 37: 676–694

## Matrix computations exhibiting sparsity under domain transformation

Weiguo Gao, Yingzhou Li, Sizhe Zhang, Peigang Jiang, Xudong Zhu & Youtuan Zhu

**Abstract** A novel framework comprising fast algorithms for operations on matrix clusters exhibiting sparsity under domain transformation is proposed in this paper. It delves into the analogs of fundamental arithmetic and crucial matrix operations within the transformed domain, alongside a devised splitting strategy. Implementation details are provided through exemplification with Cholesky factorization, matrix inversion, and SVD (singular value decomposition). We also investigate the viability of employing non-accurate and iterative methods within the transformed domain. Numerical experiments validate the correctness of the proposed framework and demonstrate significant acceleration for matrix clusters exhibiting pronounced sparsity and a limited number of columns. Consequently, this approach holds the potential to offer a promising solution for the next generation of wireless communication.

**Keywords** matrix cluster, domain transformation, Cholesky factorization, matrix inverse, SVD

**MSC(2020)** 65F05, 65F15, 65F99

**doi:** 10.1360/SSM-2024-0143